Faster exploration of some temporal graphs

Duncan Adamson* Vladimir Gusev† Dmitriy Malyshev‡ Viktor Zamaraev§

November 8, 2021

Abstract

A temporal graph $G = (G_1, G_2, \dots, G_T)$ is a graph represented by a sequence of T graphs over a common set of vertices, such that at the i^{th} time step only the edge set E_i is active. The temporal graph exploration problem asks for a shortest temporal walk on some temporal graph visiting every vertex. We show that temporal graphs with n vertices can be explored in $O(kn^{1.5} \log n)$ days if the underlying graph has treewidth k and in $O(n^{1.75} \log n)$ days if the underlying graph is planar. Furthermore, we show that any temporal graph whose underlying graph is a cycle with k chords can be explored in at most $6kn$ days. Finally, we demonstrate that there are temporal realisations of sub cubic planar graphs that cannot be explored faster than in $\Omega(n \log n)$ days. All these improve best known results in the literature.

1 Introduction

In many real world settings, networks are not static objects but instead have unstable connections that vary with time. Temporal graphs provide a model for such time-varying networks. Formally, a *temporal graph* $\mathcal{G}$ is a sequence $(G_1, G_2, G_3, \dots, G_T)$ of undirected graphs, called *snapshots*, that all share the same vertex set V, but whose edge sets $E_1, E_2, E_3, \dots, E_T$, respectively, may differ. The number $T + 1$ is called the *lifetime* of $\mathcal{G}$ and we refer to i, $0 \leq i \leq T$, as a *time* i or *day* i. The graph $G = (V, E_1 \cup E_2 \cup \cdots \cup E_T)$ is called the *underlying graph* of $\mathcal{G}$, and $\mathcal{G}$ is said to be a *temporal realisation* of G. A pair (e, i), where $e \in E_i$ is called a *time edge* of $\mathcal{G}$. A *temporal walk* from $v_1 \in V$ starting at time t to $v_k \in V$ is an alternating sequence of vertices and time edges $v_1, (e_1, i_1), v_1, \dots, v_{k-1}, (e_{k-1}, i_{k-1}), v_k$ such that $e_j = \{v_j, v_{j+1}\} \in E_{i_j}$ for $0 \leq j \leq k-1$ and $t \leq i_1 < i_1 < \dots < i_{k-1}$. The time $i_{k-1} + 1$ is called the *arrival time* of the walk.

Motivated by the central role of the Travelling Salesman problem in the world of static graphs, Michail and Spirakis [7] introduced and initiated the study of the natural temporal analogue called the Temporal Graph Exploration problem (TEXP for brevity). The goal of TEXP is to compute a temporal walk with the *earliest* arrival time that starts in a given vertex $s \in V$ and visits (i.e., *explores*) all vertices of the temporal graph. It is often convenient to describe a construction of an exploration temporal walk as the actions of an agent that is initially located at some starting vertex s and that can in every day i either stay at its current node or move to a node u that is adjacent to its current node in E_i. In the latter case, the agent *departs* from the current vertex at time i and *arrives* at u at time $i + 1$.

The decision version of TEXP in which one has to decide if at least one exploration schedule exists in a given temporal graph from a given starting vertex is an **NP**-complete problem [7]. In fact, this decision problem remains **NP**-complete even if the underlying graph has pathwidth 2 and every snapshot is a tree [2], or even if the underlying graph is a star and the exploration has to start and end at the center of the star [1].

Michail and Spirakis [7] proved that TEXP admits no $(2 - \varepsilon)$-approximation algorithm for any $\varepsilon > 0$, unless **P**=**NP**. In other words, there is no polynomial time algorithm that outputs an exploration schedule whose arrival time is at most $(2 - \varepsilon)$ times the arrival time of an optimal exploration schedule. This was substantially strengthened by Erlebach et. al. [3] who established **NP**-hardness of $n^{1-\varepsilon}$-approximation for

*Department of Computer Science, Reykjavik University, Reykjavik, Iceland. Email: duncana@ru.is

†Leverhulme Centre for Functional Materials Design, Materials Innovation Factory and Department of Chemistry, University of Liverpool, Liverpool, UK. Email: Vladimir.Gusev@liverpool.ac.uk

‡Laboratory of Algorithms and Technologies for Network Analysis, HSE University, 136 Rodionova Ulitsa, Nizhny Novgorod, 603093, Russian Federation. Email: dsmalyshev@rambler.ru

§Department of Computer Science, University of Liverpool, Liverpool, UK. Email: viktor.zamaraev@liverpool.ac.uk

Setting	Known Bounds	Our Bounds
Cycles with k-Chords	$O(6k^2 \cdot k! \cdot (2e)^k n)$ ([8])	$O(kn)$
Treewidth-k graphs	$O(k^{1.5} n^{1.5} \log(n))$ ([3])	$O(kn^{1.5} \log(n))$
Planar graphs	$O(n^{1.8} \log(n))$ ([3])	$O(n^{1.75} \log(n))$

Table 1: Summary of the new algorithms we provide for exploring temporal graphs versus the previous best known bounds.

any $\varepsilon > 0$. In fact, the result was shown for *always-connected* temporal graphs, i.e., temporal graphs in which every snapshot is a connected graph. This connectedness assumption makes the above inapproximability result tight, because one the one hand, obviously, any exploration cannot be done faster than in $n-1$ days, and on the other hand any always-connected temporal graph can be explored in at most n^2 days [7].

The strong inapproximability result for TEXP on always-connected temporal graphs motivated the study into bounds on the length of fastest exploration schedules for such temporal graphs and the present work contributes to this line of research. For convenience, from now on, unless specified otherwise we assume that every temporal graph is always-connected and has lifetime at least n^2. In [3] Erlebach et. al. demonstrated that for some temporal graphs any exploration requires $\Omega(n^2)$ days, and thus showed that the upper bound of Michail and Spirakis [7] is asymptotically best possible. This result naturally led the investigation to consider restricted temporal graphs.

One natural way to restrict a temporal graph is to restrict its underlying graph. In this direction, Erlebach et. al. [3] showed that temporal realisations of planar graphs can be explored in $O(n^{1.8} \log n)$ days; temporal realisations of graphs of treewidth at most k can be explored in $O(k^{1.5} n^{1.5} \log n)$ days; temporal realisations of $2 \times n$ grids can be explored in $O(n \log^3 n)$ days. They also showed that temporal realisations of a cycle or a cycle with a single chord can be explored in $O(n)$ days, and conjectured that any temporal graph whose underlying graph is a cycle with at most k chords can be explored in no more than $f(k) \cdot n$ days, where $f(k)$ is some function. This conjecture was recently proved by Alamouti [8] with a factorial-type function $f(k) = k^2 k! e^k$. In [4] Erlebach et. al. proved that any temporal graph in which every snapshot is a bounded-degree graph (in particular, temporal realisations of bounded-degree graphs) can be explored in $O(n^{1.75})$ days. On the negative side, Erlebach et. al. [3] constructed temporal realisations of planar graphs of degree at most 4 that cannot be explored faster than in $\Omega(n \log n)$ days.

In Erlebach and Spooner [5] considered TEXP under another natural restriction on the input temporal graphs. Namely, they studied TEXP on *k-edge-deficient* temporal graphs, i.e., temporal graphs in which every snapshot is obtained from the underlying graph by removing at most k edges. They showed that k-edge-deficient and 1-edge-deficient temporal graphs can be explored in $O(kn \log n)$ and $O(n)$ days, respectively, and constructed k-edge-deficient temporal graphs that cannot be explored faster than in $\Omega(n \log k)$ days.

Our contribution. In this work we improve a number of bounds on exploration of temporal graphs with underlying graphs from restricted classes of graphs. Table 1 provides a summary of these results and how they compare to the best known current bounds. First, in Section 2 we show that a temporal realisation of a cycle with k chords can be explored in at most $6kn$ days. Next, in Section 3 we first strengthen the exploration bound of Erlebach et. al. [3] for temporal realisations of graphs admitting a (r, b)-division; then using this bound we prove that any temporal realisation of a planar graph can be explored in $O(n^{1.75} \log n)$ days and any temporal realisation of a graph of treewidth at most k can be explored in $O(kn^{1.5} \log n)$ days. Finally, in Section 4 we demonstrate that there are temporal realisations of planar graphs of degree at most 3 that cannot be explored faster than $\Omega(n \log n)$. The latter result is tight in the sense that temporal realisations of graphs of degree at most 2 are explorable in $O(n)$ days.

Notation and tools. For a vertex set $W \subseteq V$ we denote by $\mathcal{G}[W]$ the *temporal subgraph of $\mathcal{G}$ induced by W*, i.e., the temporal graph $(G_1[W], G_2[W], G_3[W], \ldots, G_T[W])$, where for a static graph G the notation $G[W]$ means the subgraph of G induced by W. In the our proofs we will employ two useful lemmas from [3].

Lemma 1.1 (reachability, [3])**.** *Let $\mathcal{G}$ be a (not necessarily always-connected) temporal graph with a vertex set V. Let $U \subseteq V$ be a set of vertices of $\mathcal{G}$ of size k, and let $u, v \in U$. If there exists a set of $k-1$ snapshots* *each of which has a path from u to v that contains only vertices from U, then an agent can reach v starting from u and moving only in these $k-1$ snapshots.*

Notice that a straightforward consequence of the above lemma is that a temporal graph can always be explored in at most n^2 days by visiting vertices in an arbitrary order and spending at most n days to move from one vertex to the next.

The following lemma is a general reduction that transforms a multi-agent exploration schedule to a single-agent one. In the multi-agent setting, there are several agents that all start at the same vertex and move or stay put in every day independently from each other. Similarly to the single-agent setting, the goal is to visit (explore) every vertex of the temporal graph by at least one agent as soon as possible.

Lemma 1.2 (multi-agent to single-agent, [3]). *Let G be a graph on n vertices. If any temporal realisation of G can be explored in t days with k agents, then any temporal realisation of G can be explored in $O((t+n)k\log n)$ days with one agent.*

2 Cycles with bounded number of chords

Erlebach et al. proved in [3] that a temporal realisation of a cycle can be explored in at most $2n-2$ days, and this is a tight bound in the sense that there exist temporal realisations of cycles on n vertices for which any optimal exploration requires at least $2n-3$ days. In the same work it was further shown that any temporal realisation of a cycle with one chord can be explored in at most $7n$ days. Furthermore, the authors conjectured that temporal realisations of cycles with a constant number of chords are explorable in $O(n)$ days. This conjecture was confirmed in [8] where the author has shown that a temporal realisation of a cycle with k chords can be explored in at most $6k^2 \cdot k! \cdot (2e)^k n$ days. In the present section we strengthen this result by showing that any temporal realisation of a cycle with k chords can be explored in $6kn$ days. We start with the following auxiliary lemma.

Lemma 2.1. *Let $\mathcal{G} = (G_1, G_2, \ldots, G_{2n})$ be an n-vertex temporal graph of lifetime $T = 2n$, and let G be the underlying graph of $\mathcal{G}$. Let $P = (v_1, v_2, \ldots, v_k)$, $k \geq 1$, be a path in G such that every vertex of P, except possibly its endpoints v_1 and v_k, has degree 2 in G. Moreover, in every snapshot of $\mathcal{G}$ at most one edge of P is absent. Then there exists a vertex $v \in V(G)$ such that all vertices of P can be explored starting from v.*

Proof. If there exists n snapshots in $\mathcal{G}$, in which all edges of P are present, then clearly the vertices of P can be explored in this n snapshots starting from any of the endpoints of P. It can therefore be assumed that there are less than n such snapshots, i.e., there are at least $n+1$ snapshots in which exactly one edge of P is absent. We can therefore assume without loss of generality that the first $n+1$ snapshots $G_1, G_2, \ldots, G_{n+1}$ of $\mathcal{G}$ miss exactly one edge.

Observe that as every snapshot is connected and exactly one edge of P is absent in every snapshot, for every $i \in \{1, 2, \ldots, n+1\}$ the graph $H_i = G_i - \{v_1, \ldots, v_{k-1}\}$ is connected. Therefore, by Lemma 1.1, in the temporal graph $\mathcal{H} = (H_1, H_2, \ldots, H_{n+1})$ every vertex can reach any other vertex in at most $n-(k-1)-1 = n-k$ days. For every $i \in \{1, 2, \ldots, k\}$, let J_i denote some fixed temporal walk from v_1 to v_k in $\mathcal{H}$ that starts at time i and has the earliest arrival time. Note that the arrival time of J_i is at most $n-k+i$.

Now assume there are $n+1$ agents $a_1, a_2, \ldots, a_{n+1}$ that are initially placed at the vertices of $\mathcal{G}$ as follows: agent a_i is located at v_i for every $i \in \{0, 1, \ldots, k-1\}$ and all the other $n-k+1$ agents $a_k, a_{k+1}, a_{k+2}, \ldots, a_n$ are located at vertex v_k. Every day each agent will either move or stay at its current vertex. To describe the movement rules, let the *score* $\mu_t(a)$ of an agent a at time t as equal to the number vertices of P that a visited by time t. In particular, $\mu_2(a_i) = 1$ for every $i \in \{1, 2, \ldots, n+1\}$. Now, if the score $\mu_t(a)$ of an agent a at time t is $k+1$, then a does not move. Otherwise the movement of the agent a at day t is determined according to the following rules:

1. *a is at vertex v_k at time t.* If $\mu_t(a)$ is the minimum among all agents that are currently at v_k, then a moves to v_{k-1}. Otherwise a stays at v_k. If there are multiple agents with the minimum value of $\mu_t(a)$, then only the agent with the minimum index moves and all other stay at v_k. If there is no edge between v_k and v_{k-1} at time t, then the moving agent dies;

2. *a is at vertex v_i at time t, for some $i \in \{1, 2, 3, \ldots, k-1\}$.* Then a moves to v_{i-1}. As before, if there is no edge between v_i and v_{i-1} at time t, then a dies;

3. *a is at vertex* v_1 *at time* t. Then starting from time t the agent a moves according to the temporal walk J_t.

Observe that at every day at most one agent can die, and therefore after n days at least one of the agents survives. This leaves the problem of showing that any such agent has visited all vertices of P. To this end, let a_i be an agent that is alive after n days.

If $i \geq k$, then, according to the initial positions and the moving rules, a_i will start moving from vertex v_k at day $i - k$, and will visit one new vertex of P every day. Since $i \leq n$, after $i - k + k \leq n$ days a_i visits all vertices of P and stops moving.

Suppose now that $i < k$. According to the rules, after the first i days, the agent a_i moves along the path P and visits the vertices $v_i, v_{i-1}, \ldots, v_1$. After visiting all these vertices the score $\mu_i(a_i)$ of a_i at time i is equal to $i+1$ and the agent continues to move following the temporal temporal walk J_i. Let t^* be the time when a_i arrives at v_k. Observe that $\mu_{t^*}(a_i) = i + 2$ and $t^* \leq n - k + i$. By the end of day t^*, there could be at most $n - k - t^* + i$ agents at vertex v_k with a smaller score than the score $\mu_{t^*}(a_i)$ of a_i: at most $n - k + 1 - (t^* + 1) = n - k - t^*$ agents that were initially located at v_k and have not departed until the end of day t^* and at most i agents $a_i, a_{i-1}, \ldots, a_1$ that arrived at v_k earlier or at the same time as a_i (and have not departed until the end of day t^*). All other agents at v_k at time t^*, if any, have larger scores. Hence, the agent a_i will depart from v_k no later than on day

$$t^* + (n - k - t^* + i) + 1 = n - k + i + 1 = n - (k - i - 1),$$

and therefore it will survive for further $k - i - 1$ days thus visiting the remaining $k - i - 1$ vertices $v_{k-1}, v_{k-2}, \ldots, v_{i+1}$ of P. ☐

Theorem 2.2. *A temporal realisation of a cycle with* n *vertices and* k *chords can be explored in at most* $6kn$ *days.*

Proof. Let $\mathcal{G}$ be a temporal realisation of an n-vertex cycle with k chords and let G be the underlying graph of $\mathcal{G}$. Let us denote by C the underlying cycle of $\mathcal{G}$, and let $a_1, a_2, \ldots, a_s$, $s \leq 2k$, be the distinct vertices of C, ordered according to their clockwise appearance on the cycle, which are incident with at least one of the chords. For every $i \in \{1, 2, \ldots, s\}$, let P_i be the subpath of C that one obtains by following the cycle clockwise starting at a_i and ending at a_{i+1}, where the summation is modulo s.

Let $i \in \{1, 2, \ldots, s\}$ be an arbitrary fixed index. Note that all internal vertices of P_i have degree 2 in the underlying graph G. This together with the connectivity of the snapshots of $\mathcal{G}$ imply that at every day at most one edge of P_i is absent. Therefore, P_i and the temporal graph obtained by restricting $\mathcal{G}$ to any sequence of $2n$ consecutive snapshots satisfy the assumptions of Lemma 2.1. Hence, the vertices of P_i can be visited during any sequence of $3n - 1$ consecutive snapshots: Using Lemma 2.1, in the first $n - 1$ snapshots we reach a vertex v guaranteed by Lemma 2.1, and in the subsequent $2n$ snapshots, by Lemma 2.1, we visit all the vertices of P_i starting from v. Since the index i was chosen arbitrarily, the above procedure can be repeated for each of the s paths, which implies that $\mathcal{G}$ be explored in at most $2k(3n - 1) < 6kn$ days. ☐

3 Underlying graphs with (r, b)-divisions

In [3] Erlebach et al. showed that any temporal realisation of an n-vertex graph of treewidth k can be explored in $O(k^{1.5}n^{1.5}\log n)$ days, and any temporal realisation of an n-vertex planar graph can be explored in $O(n^{1.8}\log n)$ days. The key ingredient in the proofs of both results was the following

Theorem 3.1 (Theorem 4.3, [3])**.** *A temporal graph* $\mathcal{G}$*, whose underlying graph has a* (r, b)*-division*[1]*, can be explored in* $O\left((n + r^2 b)\frac{nb}{r}\log n\right)$ *days.*

In Section 3.1 we obtain a stronger version of the above theorem, which we apply in Section 3.2 to improve the exploration bounds for temporal realisations of graphs of treewidth k and planar graphs. The main technical contribution that allows us to strengthen Theorem 3.1 is Lemma 3.2 saying that if two vertex sets S and U in an n-vertex temporal graph $\mathcal{G}$ are such that $|U| \leq |S|$ and in every snapshot of $\mathcal{G}$ for every vertex $u \in U$ there exists a path between u and a vertex in S, then $|S|$ agents starting at the vertices of S (one agent per vertex) can explore vertices in U and return to their original positions in at most $4|S|n$ days.

[1]The notion of (r, b)-division is formally defined in Section 3.1

3.1 Tools

For a graph $G = (V, E)$, we say that a *vertex $v \in V$ is reachable from a vertex $u \in V$ in G* if there is a path from u to v in G. We also say that a *subset $S \subseteq V$ reaches a subset $U \subseteq V$ in G*, if every vertex $u \in U$ is reachable from some vertex in S. For a temporal graph $\mathcal{G}$ and subsets S and U of its vertices, we say that S *always reaches U in $\mathcal{G}$*, if S reaches U in every snapshot of $\mathcal{G}$.

Lemma 3.2. *Let $\mathcal{G}$ be a not necessarily always-connected temporal graph with vertex set V, let S be a subset of V of cardinality s, and let U be a subset of V with $|U| \leq s$. If S always reaches U in $\mathcal{G}$ and the lifetime of $\mathcal{G}$ is at least $4sn$, then s agents starting at the vertices of S (one agent per vertex) can explore vertices in U and return to their original positions.*

Proof. Let $S = \{x_1, x_2, \dots, x_s\}$ and let $a_1, a_2, \dots, a_s$ denote the agents that are initially located at the vertices $x_1, x_2, \dots, x_s$ respectively. For convenience, we assume that every vertex in U holds a token, and we restrict our consideration only to the exploration schedules in which the agents collect all the tokens from the vertices in U and bring them to the agents' original locations. We assume that every agent can carry at most one token at a time. We say that a vertex $u \in U$ is *explored by an agent a*, if a starts at its original position, visits u, takes the token of u, moves back to its original location, where she drops the token. A day on which agent a returns to its original location and drops the token of u will be called a *return day of a*. The assumption that an agent can carry only one token at a time implies that the agent can explore at most one vertex between any two consecutive visits to its original location.

Let $U = \{u_1, u_2, \dots, u_r\}$ and, for every $i \in [r]$, denote by t_i the earliest day by which the agents can explore i vertices in U. We will prove by induction on i that $t_i \leq 2n(s + i - 1)$. As $r \leq s$, the inequality for $i = r$ will imply the lemma.

For the base case $i = 1$, we need to show that at least one vertex in U can be explored by day $2ns$. Since every vertex in U is reachable from a vertex in S in every snapshot, by the pigeonhole principle, vertex u_1 is reachable from some fixed vertex $x \in S$ in at least $2n$ snapshots out of the first $2ns$ snapshots. Hence, by Lemma 1.1, the agent of x can explore u_1 by day $2ns$.

Let now $1 < i \leq r$ and assume that the agents can explore $i - 1$ vertices by time $t_{i-1} \leq 2n(s + i - 2)$. Suppose, towards a contradiction, that the agents cannot explore i vertices in the first $2n(s + i - 1)$ days. Let us fix a fastest exploration schedule in which the agents explore $i - 1$ vertices in U. Without loss of generality, assume that the vertices explored under this schedule are $u_1, u_2, \dots, u_{i-1}$. For $k \in [s]$, let ℓ_k be the number of vertices explored by agent a_k in the first $2n(s+i-1)$ days; note that $\ell_1 + \ell_2 + \ldots + \ell_s = i - 1$. Furthermore, we denote by $d_1^{(k)} < d_2^{(k)} < \ldots < d_{\ell_k}^{(k)}$ the return days of agent a_k and call $(d_1^{(k)}, d_2^{(k)}, \dots, d_{\ell_k}^{(k)})$ the *vector of return days of a_k*. We also assume, without loss of generality, that the schedule is *minimal* in the sense that there is no schedule in which all the agents explore the same number of vertices in U, and all the agents have the same vectors of return days, except one of the agents, say a_k, that has a vector of return days that is lexicographically smaller than $(d_1^{(k)}, d_2^{(k)}, \dots, d_{\ell_k}^{(k)})$.

Next, for an arbitrary but fixed $k \in [s]$, we will count the number of snapshots in the first $2n(s + i - 1)$ days in which vertex u_i is reachable from vertex x_k. We claim that in each of the time intervals $[1, d_1^{(k)} - 1]$, $[d_j^{(k)} + 1, d_{j+1}^{(k)} - 1], j \in [\ell_k - 1]$, $[d_{\ell_k}^{(k)} + 1, 2n(s + i - 1)]$ there are at most $2(n - 1)$ such snapshots. Indeed, if the interval $[1, d_1^{(k)} - 1]$ or any of the intervals $[d_j^{(k)} + 1, d_{j+1}^{(k)} - 1], j \in [\ell_k - 1]$ would contain $2(n - 1)$ snapshots in which vertex u_i is reachable from vertex x_k, then we could amend the schedule of agent a_k by ordering her to explore u_i during this time interval and keeping the schedule the same in the other intervals. This would produce a schedule in which a_k would have a lexicographically smaller vector of return days than $(d_1^{(k)}, d_2^{(k)}, \dots, d_{\ell_k}^{(k)})$, contradicting the minimality of the schedule. Also, if the last interval $[d_{\ell_k}^{(k)} + 1, 2n(s+i-1)]$ would contain $2(n-1)$ snapshots in which vertex u_i is reachable from vertex x_k, then agent a_k could explored u_i in this interval, contradicting the assumption that the agents cannot explore i vertices in the first $2n(s + i - 1)$ days. Hence the total number of snapshot in which u_i is reachable from vertex x_k in the first $2n(s+i-1)$ snapshots is at most $2(n-1)(\ell_k+1)+\ell_k$. Consequently, the total number of snapshots in which u_i is reachable from any vertex in S in the first $2n(s + i - 1)$ snapshots is at most

$$\sum_{k=1}^{s} \big(2(n-1)(\ell_k+1) + \ell_k\big) = 2(n-1)(s+i-1) + i - 1 < 2n(s+i-1),$$

which contradicts the assumption that S always reaches U in $\mathcal{G}$. $\square$

We will now use Lemma 3.2 to prove a stronger version of Theorem 3.1. The notion of (r,b)-division was introduced by Erlebach et al. [3] and it generalizes the notion of r-divisions used by Frederickson [6]. For positive integers r and b (which might be functions of n), a *(r,b)-division of a graph $G=(V,E)$* with n vertices is given by a set $S \subseteq V$ and a partition of $G[V \setminus S]$ into $O(n/r)$ (not necessarily connected) *components*, each associated with a *boundary set* consisting of vertices from S, such that the following properties hold:

(1) Each component contains at most r vertices.

(2) The boundary set of each component has size at most b.

(3) The boundary sets of different components may overlap, and the union of the boundary sets of all components is S.

(4) Every edge of G that has only one endpoint in a component has its other endpoint in the boundary set of that component.

Theorem 3.3. *A temporal graph $\mathcal{G}$, whose underlying graph has a (r,b)-division, can be explored in $O\big((n+\max\{r,b\}(r+b))\frac{nb}{r}\log n\big)$ days.*

Proof. We will use b agents to explore all $O(n/r)$ components one by one. Consider the exploration of a component C and its boundary set B. Since the graph is always-connected, the definition of (r,b)-division implies that B always reaches C in $\mathcal{G}[B \cup C]$, which allows us to apply Lemma 3.2 as follows. First, using Lemma 1.1, we position at most b agents at the boundary vertices in at most $n-1$ days. Next, we partition $|C|$ into $\lfloor |C|/|B| \rfloor$ subsets, each with $|B|$ elements, and the subset of the $|C| - |B| \lfloor |C|/|B| \rfloor$ remaining elements. By Lemma 3.2, any of these subsets can be explored in $4|B|(|C|+|B|)$ days in $\mathcal{G}[B \cup C]$. Therefore, C can be explored in $O\left((\lfloor |C|/|B| \rfloor + 1)|B|(|C|+|B|)\right) = O(\max\{r,b\}(r+b))$ days, and the set $B \cup C$ in $O(n+\max\{r,b\}(r+b))$ days. Consequently, the entire graph $\mathcal{G}$ can be explored in $O((n+\max\{r,b\}(r+b))\frac{n}{r})$ days using b agents, and hence, by Lemma 1.2, it can be explored in $O((n+\max\{r,b\}(r+b))\frac{nb}{r}\log n)$ days with a single agent. $\square$

3.2 Applications

3.2.1 Bounded treewidth graphs

It was shown in [3] that temporal graphs whose underlying graph has treewidth at most k can be explored in $O(k^{1.5}n^{1.5}\log n)$ days. This bound provides an improvement over the general $O(n^2)$ bound whenever $k = o\left(n^{1/3}/\log^{2/3} n\right)$. A key ingredient of the proof of this result was the fact that graphs with treewidth at most k admit a $\left(2\sqrt{n/k}, 6k\right)$-division (see Lemma 4.4 in [3]). Using exactly the same proof as in [3], but replacing $\sqrt{\frac{n}{k}}$ and $\sqrt{nk}$ with $\sqrt{n}$ everywhere, one can obtain the following

Lemma 3.4 (adaptation of Lemma 4.4 [3])**.** *Any graph of treewidth at most k admits a $\left(2\sqrt{n}, 6k\right)$-division.*

We will use this latter fact together with Theorem 3.3 to derive an improved exploration bound for graphs of treewidth at most k.

Theorem 3.5. *An n-vertex temporal graph, whose underlying graph has treewidth at most k, can be explored in $O(kn^{1.5}\log n)$ days.*

Proof. We can assume, without loss of generality, that $k = o(n^{0.5})$, as otherwise the bound in the statement becomes $\omega(n^2)$ and the result clearly holds. Under this assumption, Lemma 3.4 and Theorem 3.3 imply that an n-vertex temporal graph, whose underlying graph has treewidth at most k, can be explored in $O\left((n+\sqrt{n}(\sqrt{n}+k))\sqrt{n}k\log n\right) = O(kn^{1.5}\log n)$ days. $\square$

We note that Theorem 3.5 improves the previous bound from [3] as well as implies an improvement over the general $O(n^2)$ bound for graphs with treewidth $k = o\left(n^{0.5}/\log n\right)$.

3.2.2 Planar graphs

It was shown in [3] that temporal realisations of planar graphs can be explored in $O(n^{1.8}\log n)$ days. We will follow a similar strategy as in [3] but use our Theorem 3.3 to reduce the bound to $O(n^{1.75}\log n)$.

Theorem 3.6. *An n-vertex temporal graph, whose underlying graph is planar, can be explored in $O(n^{1.75}\log n)$ days.*

Proof. Frederickson proved that planar graphs admit $(r, O(\sqrt{r}))$-divisions for any $1 \le r \le n$ [6]. Applying this result with $r = \sqrt{n}$ and Theorem 3.3 we conclude that a temporal realisation of an n-vertex planar graph can be explored in $O\left((n + r(r+\sqrt{r}))\,\frac{n}{\sqrt{r}}\log n\right) = O\left((n^2/\sqrt{r} + r^{1.5}n)\log n\right) = O\left(n^{1.75}\log n\right)$ days. □

4 Subcubic planar graphs

Temporal realisations of graphs of maximum degree at most 2 can be explored in linear time. Indeed, a connected graph of maximum degree 2 is either a path or a cycle. Temporal realisations of paths are trivially explorable in linear time, because every snapshot in such temporal graphs must be the same connected graph. As shown in [3], temporal realisations of cycles are also explorable in linear time. On the other hand, it was proved in [3] that some temporal realisations of planar graphs of maximum degree 4 cannot be explored faster than in $\Omega(n\log n)$ days even if every snapshot is a path.

Theorem 4.1 (Theorem 4.1, [3])**.** *There exist temporal realisations of n-vertex planar graphs of maximum degree 4, in which every snapshot is a path, that cannot be explored faster than $\Omega(n\log n)$ days.*

This leaves an intriguing open case of exploration time of temporal realisations of planar graphs of maximum degree 3. In particular, are such temporal graphs always explorable in linear time? As we shall see below, in general, such temporal graphs can require $\Omega(n\log n)$ days for their exploration. To prove the result, we will transform the construction from the proof of Theorem 4.1 and apply the following edge contraction lemma from [3].

Lemma 4.2 (edge contraction, Lemma 2.4, [3])**.** *Let G be a graph such that every temporal realisation of G with lifetime at least t can be explored in t days. Let G' be a graph that is obtained from G by contracting edges. Then every temporal realisation of G' with lifetime t can also be explored in t days.*

Theorem 4.3. *There exist temporal realisations of n-vertex* subcubic *planar graphs that cannot be explored faster than $\Omega(n\log n)$ days.*

Proof. Let $n' \ge 16$, let G be an n'-vertex planar graph of maximum degree 4, and let $\mathcal{G}$ be a temporal realisation of G for which every exploration requires at least $cn'\log n'$ days for some positive constant c. Such G and $\mathcal{G}$ exist by Theorem 4.1.

From graph G we obtain a graph H as follows. For every vertex u in G with 4 neighbours v_1, v_2, v_3, v_4, we delete u from G, add 4 new vertices u_1, u_2, u_3, u_4, forming a 4-cycle (u_1, u_2, u_3, u_4), and add 4 new edges $\{u_1, v_1\}$, $\{u_2, v_2\}$, $\{u_3, v_3\}$, $\{u_4, v_4\}$ (see Fig. 1). Let n be the number of vertices in H. Clearly, $n' \le n \le 4n'$ and G is obtained from H by contracting edges. Furthermore, it is not hard to see that H is planar and every vertex in H has degree at most 3. Therefore, there exists a temporal realisation $\mathcal{H}$ of H for which every exploration requires at least $cn'\log n'$ days. Indeed, otherwise, by Lemma 4.2, $\mathcal{G}$ could be explored in less than $cn'\log n'$ days, which would contradict our assumption. Hence, $\mathcal{H}$ cannot be explored in less than

$$cn'\log n' \ge c\frac{n}{4}\log\frac{n}{4} \ge \frac{c}{8}n\log n = \Omega(n\log n)$$

days, where the latter inequality uses the assumption that $n \ge 16$. □

Remark. We note that in the temporal graph $\mathcal{G}$ from the proof of Theorem 4.1 every snapshot is a path. The transformation in the above proof of Theorem 4.3 can be easily specified in such a way that every snapshot in $\mathcal{H}$ is also a path. We leave the details to the interested reader.

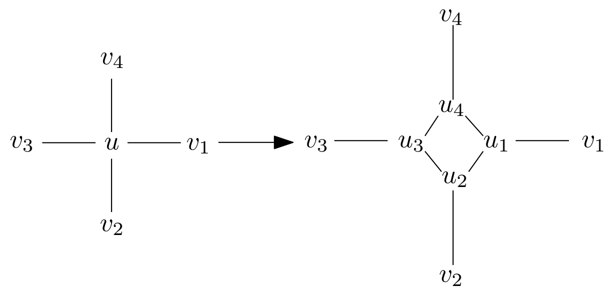

Figure 1: An illustration of the transformation from the neighbourhood around the vertex v in G to the 4-cycle (u_1, u_2, u_3, u_4) in H.

Acknowledgments

The work of Dmitriy Malyshev was conducted within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE).

References

[1] Eleni C Akrida, George B Mertzios, Paul G Spirakis, and Christoforos Raptopoulos. The temporal explorer who returns to the base. *Journal of Computer and System Sciences*, 120:179–193, 2021.

[2] Hans L Bodlaender and Tom C van der Zanden. On exploring always-connected temporal graphs of small pathwidth. *Information Processing Letters*, 142:68–71, 2019.

[3] Thomas Erlebach, Michael Hoffmann, and Michael Kammer. On temporal graph exploration. *Journal of Computer and System Sciences*, 119:1–18, 2021.

[4] Thomas Erlebach, Frank Kammer, Kelin Luo, Andrej Sajenko, and Jakob T Spooner. Two moves per time step make a difference. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[5] Thomas Erlebach and Jakob T. Spooner. Exploration of k-edge-deficient temporal graphs. In Anna Lubiw and Mohammad Salavatipour, editors, *Algorithms and Data Structures*, pages 371–384, Cham, 2021. Springer International Publishing.

[6] Greg Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal of Computing*, 16:1004–1022, 1987.

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

[8] Shadi Taghian Alamouti. Exploring temporal cycles and grids, 2020.