Non-Strict Temporal Exploration

Thomas Erlebach[0000−0002−4470−5868] and Jakob T. Spooner[0000−0003−3816−6308]

School of Informatics, University of Leicester, Leicester, England

Abstract. A *temporal graph* $\mathcal{G} = \langle G_1, ..., G_L \rangle$ is a sequence of graphs $G_i \subseteq G$, for some given *underlying graph* G of order n. We consider the *non-strict* variant of the Temporal Exploration problem, in which we are asked to decide if $\mathcal{G}$ admits a sequence W of consecutively crossed edges $e \in G$, such that W visits all vertices at least once and that each $e \in W$ is crossed at a *timestep* $t' \in [L]$ such that $t' \geq t$, where t is the timestep during which the previous edge was crossed. This variant of the problem is shown to be NP-complete. We also consider the hardness of approximating the exploration time for yes-instances in which our order-n input graph satisfies certain assumptions that ensure exploration schedules always exist. The first is that each pair of vertices are contained in the same component at least once in every period of n steps, whilst the second is that the temporal diameter of our input graph is bounded by a constant c. For the latter of these two assumptions we show $O(n^{\frac{1}{2}-\varepsilon})$-inapproximability and $O(n^{1-\varepsilon})$-inapproximability in the $c = 2$ and $c \geq 3$ cases, respectively. For graphs with temporal diameter $c = 2$, we also prove an $O(\sqrt{n} \log n)$ upper bound on worst-case time required for exploration, as well as an $\Omega(\sqrt{n})$ lower bound.

1 Introduction

Given a connected, undirected graph G of order n, an $O(n)$ upper bound on the length of a minimal walk that *explores* G (i.e., visits in an arbitrary order, all $v \in V(G)$ at least once) can be easily obtained by considering the length of an Euler tour around a spanning tree of G. The situation is altered considerably if we allow for the edge-set of the graphs in our input space to change over the course of some discretised time period, assuming that the vertex set remains constant at each point in this period. Such graphs have in recent years been referred to as *temporal, dynamic* or *time-varying*, and indeed it is known that there exist infinitely many graphs $\mathcal{G}$ of this sort that are connected at each point in time, and such that their exploration requires $\Omega(n^2)$ moves (where a move can consist of traversing an edge, or waiting at the current vertex), where $n = |V(\mathcal{G})|$ [8]. Due in large part to the frequency at which highly dynamic networks arise in the modelling of practical, real-life situations, an effort to better understand temporal graph models, along with the various optimisation problems defined upon them (e.g., the exploration problem considered here), has been made in recent years. For a more detailed introduction to the concept of temporal graphs and related combinatorial problems the reader is referred to [17].

Much of the existing work regarding temporal graph exploration sees a temporal graph defined as a length-L sequence of static graphs $\mathcal{G} = \langle G_1, ..., G_L \rangle$, where each G_i has the same vertex set as some given *underlying graph* G, but can have an edge set that is a proper subset of $E(G)$. The Temporal Exploration problem (TEXP for short) then asks that, given a temporal graph $\mathcal{G}$ and some prespecified start vertex $s \in V(\mathcal{G})$, we compute a *foremost exploration schedule* starting from s – a sequence of edges crossed during strictly increasing timesteps (equivalently, at most one edge can be crossed per timestep), such that all vertices are visited at least once and that the timestep in which the last unvisited vertex is reached is minimal.

In this paper, we relax the condition that edges in a feasible exploration schedule must be crossed during strictly-increasing timesteps, and allow for any number of edges to be crossed in each step. Such a scenario arises for example in delay-tolerant networks [20]. Such networks tend to be disconnected at any time, and the speed at which the network topology changes is often much slower than the speed at which messages can be transmitted. Therefore, a mobile agent could visit any network node in its current connected component before the topology changes. It is clear that allowing for an agent to make an arbitrary number of moves across edges in a single time step alters the nature of the exploration problem considerably. In particular, it no longer makes sense to restrict our input space to always-connected graphs, since a trivial bound of a single step can be obtained by employing the same Euler tour-based technique that can be used to explore any static graph. As such, it is more natural to assume that a given input graph $\mathcal{G}$ consists of a number of disjoint components in each step. This, however, means that we cannot always guarantee that, for arbitrary $\mathcal{G}$ and start vertex $s \in V(\mathcal{G})$, $\mathcal{G}$ admits an exploration schedule starting at s. Given that we relax the requirement that edges be crossed in strictly-increasing timesteps, we dub the problem of deciding, in this model, whether or not a given temporal graph $\mathcal{G}$ admits an exploration schedule Non-Strict TEXP, showing it to be NP-complete in general. We then consider two seemingly natural assumptions regarding the connectivity of the vertices which, when satisfied by our input graph, ensure that exploration is always possible. The first of these (which we name *pairwise vertex-togetherness*) posits that every pair of vertices will be contained in the same component at least once every n steps, where n is the graph's order – we prove $O(n^{1-\epsilon})$-inapproximability in this case. The second assumption insists that every pair of vertices in our input graph are able to reach one another in at most a constant c many steps. We note that this is equivalent to insisting that our input graph have temporal diameter bounded by a constant c (using the natural adaptation of the definition of temporal diameter from [19] to the non-strict model). For the latter assumption an obvious $O(n)$ upper bound on exploration time exists, and we show that when $c \geq 3$ this bound is in fact tight. For $c = 2$, we prove upper and lower bounds of $O(\sqrt{n} \cdot \log n)$ and $\Omega(\sqrt{n})$ respectively, leaving just a $\Theta(\log n)$ factor's gap between the two. Amongst other things, we also consider the hardness of approximating optimal solutions for the cases of temporal diameter 2 and ≥ 3, and lower bounds showing

$O(n^{\frac{1}{2}-\varepsilon})$ inapproximability in the former case and $O(n^{1-\varepsilon})$ inapproximability in the latter are provided (where n is the order of the input graph).

2 Related Work

Brodén et al. [3] considered the Temporal Travelling Salesperson Problem on a graph with n vertices that is complete in every timestep, but had edge-costs which differed between 1 and 2 from step to step. Even when each edge's cost can change at most k times during the lifetime of the graph, they showed that the problem is NP-complete, but were able to provide a $(2 - \frac{2}{3k})$-approximation. Michail and Spirakis [19] showed the same problem to be APX-hard and provided an improved $(1.7 + \epsilon)$-approximation. Bui-Xuan et. al proposed in [4] a number of natural objectives to consider when computing a temporal walk/path, amongst which were *fastest* (minimum difference between departure and arrival time) and *foremost* (minimum arrival time) which is considered here. Also introduced in [19] was the Temporal Exploration problem (TEXP), by which we are asked to decide whether a given temporal graph admits an exploration schedule. They showed that this is NP-complete when no restrictions are placed on the input graph, and that even when the graph is connected in every timestep, approximating foremost exploration schedules with ratio $(2 - \epsilon)$ is NP-hard. Erlebach et al. [8] considered the Temporal Exploration problem under the *always-connected* property introduced in [19], improving the previously best-known inapproximability ratio to $O(n^{1-\epsilon})$. Complementing the aforementioned $\Omega(n^2)$ lower bound on the time needed to explore general always-connected temporal graphs they proved a $O(n^2)$ upper bound, as well as a number of subquadratic/superlinear upper/lower bounds for restricted subclasses of always-connected temporal graphs. In a similar vein, Bodlaender and van der Zanden [2] considered TEXP when the input graph has pathwidth at most 2 in every step, showing the decision variant to be NP-complete under these restrictions. In [1], Akrida et al. consider a variant of TEXP in which a candidate solution must return to the vertex from which it initially departed, focusing on the case in which the input graph has an underlying star. They gave an $O(n \log n)$-time algorithm deciding whether a given *temporal star* is explorable or not, under the restriction that each edge of the star is present in at most 2 or 3 time steps. In [15] and [14], the problem of temporal exploration is considered on the classes of temporal graphs with underlying cycles and cactuses, respectively. In [9], the authors prove an $O(dn^{1.75})$ bound on the number of time steps required to explore any temporal graph with degree bounded by d in each step, a considerable improvement over the previously best known $O(\frac{n^2 \log d}{\log n})$ bound [10]. Interestingly, the same bound can also be extended to general always-connected graphs when restrictions are relaxed and a computed exploration schedule is allowed to cross two edges in any given timestep – this is owed to the fact that the square of any static graph G admits a bounded-degree spanning tree. Notions of strict/non-strict paths which respectively allow for a single/infinitely many edge(s) to be crossed in any given timestep have been considered before, notably by Kempe et al. in [16] and Fluschnik et al. in [12].

Various other studies related to variants of exploration/path problems in temporal graphs have been considered. For example, the authors in [13] and [7] consider the problem of temporal exploration from a distributed standpoint. Casteigts et al. [6] consider a variant of the problem of finding a path between a given pair of vertices s and t, in which there is an upper bound on the number of timesteps that a computed path P is allowed to wait at any vertex $v \in P$ before crossing the next edge. For a more comprehensive overview of temporal graph problems and the various temporal graph classes on which they may be defined, the reader is referred to [17, 5].

3 Graph Model and Problem Definition

Throughout the following denote by $[n]$ the set of integers $\{1, 2, ..., n\}$, and by $[x, n]$ $(x < n)$ the set of integers $\{x, x + 1, ..., n\}$. A standard way of defining a temporal graph within the literature is as a length-L sequence of graphs $\mathcal{G} = \langle G_1, ..., G_L \rangle$. Here L is the *lifetime* of the graph, and we require that for every $i \in [L]$, G_i is a subgraph of the *underlying graph* G of $\mathcal{G}$. In particular, we have that $V(G_i) = V(G)$ and $E(G_i) \subseteq E(G)$ for all $i \in [L]$.

As was previously noted, in the context of the non-strict variant of TEXP, it no longer makes sense to restrict our attention to the class of always-connected graphs. Therefore, we assume that for a given temporal graph $\mathcal{G} = \langle G_1, ..., G_L \rangle$, G_i $(i \in [L])$ consists of some number ≥ 1 of distinct connected components. Moreover, since any number of edges can be crossed in a given step, the edge structure of each component no longer matters – all that is important when attempting to compute an exploration schedule W is which component $C \in G_i$ is occupied by W in timestep i, since all $v \in C$ can be explored during that step. We can therefore, without loss of generality, use the following definition of a *non-strict temporal graph*:

Definition 1 (Non-strict temporal graph, $\mathcal{G}$). *A non-strict temporal graph $\mathcal{G} = \langle G_1, ..., G_L \rangle$ with vertex set $V := V(\mathcal{G})$ and lifetime L is an indexed sequence of partitions $G_i = \{C_{i,1}, ..., C_{i,s_i}\}$, with $i \in [L]$. For all $i \in [L]$, every $v \in V(G)$ satisfies $v \in C_{i,j_i}$ for a unique $j_i \in [s_i]$.*

Definition 2 (Non-strict temporal walk, W). *A length-k non-strict temporal walk $W = C_{1,j_1}, C_{2,j_2}, ..., C_{k,j_k}$ through a non-strict temporal graph $\mathcal{G} = \langle G_1, ..., G_L \rangle$ is a sequence of components C_{i,j_i} such that, for all $i \in [k]$, $C_{i,j_i} \in G_i$ and $j_i \in [s_i]$. Additionally, $k \in [L]$, where L is the lifetime of the graph upon which W is defined. We also require that $C_{i,j_i} \cap C_{i+1,j_{i+1}} \neq \emptyset$ for all $i \in [k-1]$, so that it is ensured that the $(i+1)$-th component visited by W can be reached from the i-th component if and only if W ends step t_i at a vertex that lies in the intersection of these two components. For all $i \in [k]$ we say W visits all $v \in C_{i,j_i}$.*

A non-strict temporal walk $W = C_{1,j_1}, ..., C_{k,j_k}$ around a given graph $\mathcal{G}$ is an *exploration schedule* if and only if, for all $v \in V(\mathcal{G})$, there exists an $i \in [k]$ such that $v \in C_{i,j_i}$. Throughout the remainder of the paper, we may refer to non-strict temporal graphs and non-strict temporal walks simply as graphs and walks, respectively. Further, we might speak in terms of a *mobile agent* (or *agent*) which we assume to be, at any timestep, following a non-strict temporal walk around a given non-strict temporal graph in an attempt to explore it. We define the decision version of the NON-STRICT TEMPORAL EXPLORATION problem as follows:

Definition 3 (Non-Strict Temporal Exploration). *An instance of the* NON-STRICT TEMPORAL EXPLORATION *(*NS-TEXP*) problem is given as a tuple $(\mathcal{G}, s)$, where $\mathcal{G}$ is a given non-strict temporal graph with lifetime L and underlying graph G, and $s \in V(G)$. The problem then asks that we decide whether $\mathcal{G}$ admits an exploration schedule $W = C_{1,j_1}, ..., C_{k,j_k}$ starting from s, i.e., such that $s \in C_{1,j_1}$.*

If we consider only the set of yes-instances of NS-TEXP, i.e., those instances $(\mathcal{G}, s)$ such that $\mathcal{G}$ admits an exploration schedule starting from $s \in V(\mathcal{G})$, then it also makes sense for us to consider optimisation variants of NS-TEXP. In particular, we consider a variant FOREMOST-NON-STRICT TEXP (FNS-TEXP for short) which asks that we compute a *foremost* exploration schedule $W = C_{1,j_1}, ..., C_{k,j_k}$, i.e., one for which $k \leq l$ for any other exploration schedule $W' = C'_{1,j_1}, ..., C'_{l,j_l}$.

4 Deciding Whether Exploration Is Possible

Theorem 1. NON-STRICT TEMPORAL EXPLORATION *is* NP*-complete.*

Proof. The problem is in NP because an exploration schedule is of polynomial size (note that the input size is $\Omega(NL)$ for a temporal graph with N vertices and lifetime L, while an exploration schedule has size $O(NL)$) and its validity may be checked in polynomial time. To prove that the problem is NP-hard we give a reduction from 3SAT. Let instance I of 3SAT be given by variables $x_1, ..., x_n$ and clauses $c_1, ..., c_m$. Without loss of generality, assume that no clause contains x_i and $\bar{x}_i$ for any i. We proceed by creating a temporal graph $\mathcal{G}$ with vertex set $V(\mathcal{G}) = \{s\} \cup \{x_i^T, x_i^F : 1 \leq i \leq n\} \cup \{c_j : 1 \leq j \leq m\}$ and lifetime $2n$. The connected components of the graph in each step t are as follows (assume that all unmentioned vertices in each step are disconnected in $\mathcal{G}$): In step 1, let $\{s, x_1^T, x_1^F\}$ form one component. In every subsequent step $2i$ with $i \in [n]$, let there be a *true component* containing x_i^T and all clause vertices c_j that correspond to a clause of I which is satisfied by setting $x_i = 1$, as well as a *false component* containing x_i^F and all clause vertices c_j corresponding to clauses satisfied by setting $x_i = 0$. In all remaining steps $2i - 1$ for $i \in [2, n]$, let there be a component $\{x_{i-1}^T, x_{i-1}^F, x_i^T, x_i^F\}$. To complete the proof we show that there exists a satisfying assignment for I if and only if there exists an exploration schedule of $\mathcal{G}$.

($\implies$) Since I is satisfiable, there exists a satisfying assignment $\alpha : X \to \{0,1\}$ of boolean values to all $x_i \in X$. We claim that the following produces an exploration schedule W of $\mathcal{G}$: In the $(2i-1)$-th step ($i \in [n]$), W will be positioned in the component containing the vertices $\{x^T_{i-1}, x^F_{i-1}, x^T_i, x^F_i\}$ (as well as s in case $i = 1$). Explore both x^T_i and x^F_i, finishing at x^T_i if $\alpha(x_i) = 1$ or x^F_i is $\alpha(x_i) = 0$. At the start of step $2i$ ($i \in [n]$), W can either be in the component containing x^T_i or the one containing x^F_i (depending now on the value of $\alpha(x_i)$), and can explore all vertices c_j in that component and move back to x^T_i or x^F_i.

By definition of our reduction, the c_j explored by the produced walk W are precisely those which correspond to the clauses satisfied by assignment α. Since α is satisfying, each clause is satisfied by setting $x_i = \alpha(x_i)$ for at least one x_i; as such, W explores all vertices x^T_i, x^F_i and c_j as required.

($\impliedby$) Let $\mathcal{G}$ be the input graph of the Non-Strict TEXP instance produced by our reduction from I. Assume that $\mathcal{G}$ admits an exploration schedule W. By construction, for all $j \in [m]$, moves to and from vertices c_j can only be made during steps in which they are contained within the true/false component. Since W is an exploration schedule it must visit all c_j, and so for each j there must exist some $i \in [n]$ such that W initially reaches c_j within either the true or false component of step $2i$. Since each c_j is placed in the true/false component of step $2i$ only when $x_i = 1/x_i = 0$ satisfies the corresponding clause of I, a satisfying assignment for I can be obtained by checking, for every $i \in [n]$, whether W visits the true or false component, setting $x_i = 1$ in the former case and $x_i = 0$ in the latter. (Note that in steps $2i$ during which neither the true/false component are visited, we can choose an arbitrary setting for x_i.) $\square$

5 Exploration with Pairwise Vertex-Togetherness

We next consider instances of NS-TEXP for which the input graph $\mathcal{G}$ satisfies the following assumption, which we refer to as *pairwise vertex-togetherness*:

Assumption 1 (Pairwise vertex-togetherness) *All pairs of vertices $u, v \in V(\mathcal{G})$ are contained in the same connected component at least once during every period of N steps, where $N = |V(\mathcal{G})|$.*

The following algorithm enables us to explore any graph $\mathcal{G}$, with lifetime $L \geq N$, such that $\mathcal{G}$ satisfies Assumption 1: Start at the specified start vertex s, and in any of the steps $1 \leq i \leq N$ in which s is contained in the same connected component as some currently unexplored vertices, visit those vertices and move back to s by the end of step i. To see that this in fact produces an exploration schedule, observe that by Assumption 1 s will be contained in the same connected component as each $v \in V(\mathcal{G})$ at least once during the steps $1 \leq i \leq N$. Note that this also implies an N-approximation algorithm for the Non-Strict TEXP problem (consider the instance in which the graph in the first step consists of a single connected component). In complement to this observation, we state the following result:

Theorem 2. *Even when the input graph $\mathcal{G}$ satisfies Assumption 1, it is* NP*-hard to approximate solutions to* FOREMOST-NON-STRICT TEXP *with ratio $\Theta(N^{1-\varepsilon})$ for any $\varepsilon > 0$, where N is the order of the input graph.*

Proof. Let $NST = \langle \mathcal{G}, s \rangle$ be any instance of the NON-STRICT TEXP decision problem, obtained by performing the reduction of Theorem 1 on an instance of 3SAT with n literals and m clauses. By the reduction, the graph $\mathcal{G}$ consists of $2n$ literal vertices, x_i^T and x_i^F $(1 \leq i \leq n)$, m clause vertices c_j $(1 \leq j \leq m)$, and an additional start vertex s. To reduce to FNS-TEXP from NST, we construct an instance $FNST = \langle \mathcal{G}', s' \rangle$ as follows: Let $V(\mathcal{G}') = V(\mathcal{G}) \cup \{d_1, ..., d_{n^c}\}$, where the vertices $d_1, d_2, ..., d_{n^c}$ are n^c dummy vertices (for some constant $c > 1$). Let $N = |V(\mathcal{G}')| = 2n+m+n^c$, and let $L = N$ be the lifetime of $\mathcal{G}'$. The components of $\mathcal{G}'$ in each step of its lifetime are defined to be as follows: In step 1, the graph consists of the connected component $\{s, d_1, d_2, ..., d_{n^c}\}$, and all clause and literal vertices lie disconnected in their own components. In the steps $t \in [2, 2n+1]$, the step t components of $\mathcal{G}'$ are the same as the step $t-1$ components of $\mathcal{G}$, and we create an additional n^c components containing each of the d_i. During the steps $t \in [2n+2, N-1]$, every vertex lies in one component on its own, and then in the N-th and final step all vertices belong to one single component.

Since, during the steps $t \in [2n+2, N-1]$, all vertices are disconnected in $\mathcal{G}'$, it follows that no new vertices can be explored during any of these steps. We therefore distinguish between the following two cases, showing that deciding whether $O(n)$ time steps suffice to explore $\mathcal{G}'$ or whether $\Theta(n^c)$ timesteps are required also decides whether or not $\mathcal{G}$ admits an exploration schedule.

$\mathcal{G}'$ can be explored in $2n+1$ steps: By construction, none of the vertices c_j, x_i^T or x_i^F can be reached in $\mathcal{G}'$ from s until the start of the second step. Therefore, it must be that any exploration schedule with length $\leq 2n+1$ starting at s visits all of these vertices during the steps $t \in [2, 2n+1]$. Observe now that, by construction, the step $t \in [2, 2n+1]$ components of $\mathcal{G}'$ are the step $t-1$ components of $\mathcal{G}$, and that these steps constitute the entire lifetime of $\mathcal{G}$ – from this it follows that there must exist a valid exploration schedule of $\mathcal{G}$.

Exploring $\mathcal{G}'$ requires N steps: We claim that, since $\mathcal{G}'$ requires N steps to be explored completely, it must be that $\mathcal{G}$ admits no exploration schedule. To see this, recall once more that the step $t \in [2, 2n+1]$ components of $\mathcal{G}'$ are the step $t-1$ components of $\mathcal{G}$ (with each $t-1 \in [1, 2n]$) and so it must be that no temporal walk W starting at s in time step 1 visits all vertices by the end of time step $2n$. Otherwise, we would be in case (1) and it would have been possible to explore $\mathcal{G}'$ by the end of step $2n+1$ by visiting s and all d_i in step 1, then following an exploration schedule for $\mathcal{G}$ in $\mathcal{G}'$ during the steps $t \in [2, 2n+1]$.

Since deciding whether or not $\mathcal{G}$ can be fully explored is NP-complete, it follows from the above case analysis that it is NP-hard to approximate solutions to FOREMOST NON-STRICT TEXP instances in which $\mathcal{G}$ satisfies Assumption 1) with ratio $\frac{\Theta(n^c)}{\Theta(n)} = \Theta(N^{\frac{c-1}{c}}) = \Theta(N^{1-\varepsilon})$, where $\varepsilon = \frac{1}{c} > 0$ can be forced arbitrarily close to 0 by choosing the constant c large enough. $\square$

6 Exploration with Bounded Temporal Diameter

One further assumption which, when satisfied, ensures that complete exploration of a given temporal graph $\mathcal{G}$ is always possible (provided that the lifetime of $\mathcal{G}$ is suitably long) is the following:

Assumption 2 (Constant-bounded temporal diameter) *For every pair of vertices $u, v \in V(\mathcal{G})$, u can reach v in at most c steps (for some $c = O(1)$) and this holds from any time step t.*

An obvious upper bound on the number of steps required to fully explore any temporal graph $\mathcal{G}$ of order n such that $\mathcal{G}$ satisfies Assumption 2 (for a constant c), and $(\mathcal{G}, s)$ is a yes-instance of NS-TEXP, is $c(n-1)$: Starting from s, an agent in $\mathcal{G}$ could repeatedly select an arbitrary unexplored vertex and move to it in at most c steps, repeating this process $n-1$ times (once for each vertex $v \in V(\mathcal{G}) - \{s\}$).

6.1 Hardness of the Decision Problem for Temporal Diameter 2

The following result is concerned with the NP-completeness of deciding instances $(\mathcal{G}, s)$ of NS-TEXP in which $\mathcal{G}$ satisfies Assumption 2 for $c = 2$:

Theorem 3. *Deciding* Non-Strict Temporal Exploration *is NP-complete, even when restricted to instances in which the input graph $\mathcal{G}$ satisfies Assumption 2 for $c = 2$.*

Proof. The reduction is from the NP-complete problem of 3SAT restricted to instances in which each literal occurs in at most 4 clauses, which we dub 3SAT* [21]. Given an instance I of 3SAT* comprised of $n \geq 3$ variables and $m = O(n)$ clauses, we proceed by constructing a non-strict temporal graph $\mathcal{G}$ (with lifetime $n+3$) that satisfies the connectivity assumption for $c = 2$, such that $\mathcal{G}$ is fully explorable if and only if I is satisfiable. To do so we create 2 *literal vertices* x_i^T and x_i^F for each variable x_i of 3SAT* instance I. We then create $n+3$ *clause copy vertices* $c_{j,k}$ ($k \in [n+3]$) for all m clauses of I. Finally, we create $2(2n+m(n+3))^2$ many *connectivity vertices* v_i ($i \in [2(2n+m(n+3))^2]$) and divide them into two groups, the *red* group and the *blue* group, each of size $(2n+m(n+3))^2$. In steps 1 and 2, arrange the red connectivity vertices as a $2n+m(n+3)$ by $2n+m(n+3)$ grid, then let all rows of this grid lie in separate components during step 1, and all columns of the grid lie in separate components during step 2. Arbitrarily set the start vertex to be $s = x_1^T$. To the first row component in step 1, add the vertices x_i^T and x_i^F (for all $i \in [n]$), along with all blue connectivity vertices. Then, add each of the remaining clause vertices to a unique component (of which there are $2n + m(n+3) - 1$ remaining). In step 2, we now arrange the blue connectivity vertices as a $2n + m(n+3)$ by $2n + m(n+3)$ grid, with each of the step 2 components initially containing one red column and one blue row. Now, add each of the non-connectivity vertices (i.e., all literal and clause-copy vertices) to an arbitrary component, ensuring that no component contains more than one non-connectivity vertex (this is possible since there are $2n + m(n + 3)$ such vertices and the same number of components). In all steps $t \in [3, n+3]$, we let the blue vertices alternate between being columns and rows, so that in each step there are exactly $2n + m(n + 3)$ components. From step 3 onward, all red connectivity vertices will belong to a unique but arbitrarily selected component. All literal and clause-copy vertices should be added to one of the $2n + m(n + 3)$ components in step 3. For all steps $t \in [4, n + 3]$, add the literal vertex x_{t-3}^T to the first component, along with all clause-copies that correspond to a clause of 3SAT* instance I satisfied by setting $x_{t-3} = 1$; to the second component, add the literal vertex x_{t-3}^F alongside all clause-copies corresponding to a clause satisfied by setting $x_{t-3} = 0$. (We will from here onward refer to the 'first' and 'second' component as the *true* and *false* component.) In any of these steps, all remaining literal and clause-copy vertices should be assigned to an arbitrary but unique component that are neither the true or false component of that step. Clearly, $\mathcal{G}$ satisfies the $c = 2$ connectivity assumption, since in step 2 we have that all $2n + m(n + 3)$ components contain exactly one of the red vertices in each of the step 1 components, and for all pairs of consecutive steps i and $i + 1$ for $i > 1$, the same holds for the blue vertices. Therefore, starting in step i, it is possible to be positioned in any step $i + 1$ component (and therefore reach any vertex in at most a single step) by moving to the appropriate red/blue vertex and waiting until the start of the next step. To complete the proof, we show that I is satisfiable if and only if $\mathcal{G}$ is explorable:

($\implies$) To construct an exploration schedule of $\mathcal{G}$ from a satisfying assignment for 3SAT instance I, we can use the first three steps of $\mathcal{G}$'s lifetime in order to visit all blue/red connectivity vertices, as well as the literal vertices. For the remaining steps $t \in [4, n + 3]$, visit the true component in step i if $x_{t-3} = 1$ or the false component otherwise; this is possible due to the connectivity vertices. It is clear, by arguments similar to those used in the proof of Theorem 1, that $\mathcal{G}$ is an exploration schedule since it was constructed from a satisfying assignment for I.

($\impliedby$) First observe that no c_j can be reached until step 2. We have $n + 3$ $c_{j,k}$ associated with the j-th clause of I; since there are only $n + 2$ remaining steps it is not possible to visit all of the copies associated with the j-th clause in steps in which they are not contained in either the true or false component (i.e., one per timestep). Therefore, for all $j \in [m]$ there is at least one timestep in which ≥ 2 $c_{j,k}$ are visited whilst both contained in the true or false component, and so by construction the remaining $n + 2$ copies can also be visited during that same step. Moreover, since W is an exploration schedule all $c_{j,k} \in V(\mathcal{G})$ ($j \in [m], k \in [n + 3]$) must be visited; hence all clauses of I can be satisfied by setting variable $x_{t-3} = 1$ if W visits the true component in step $t \in [4, n + 3]$, and $x_{t-3} = 0$ otherwise (an arbitrary setting of x_{t-3} suffices when neither are visited). □

Due to the fact that any graph $\mathcal{G}$ satisfying Assumption 2 for some constant c also satisfies it for every $d > c$, we obtain as a corollary of Theorem 3 the following:

Corollary 1. *Deciding* NON-STRICT EXPLORATION *is* NP*-complete when restricted to graphs satisfying Assumption 2 for any* $c \geq 3$.

6.2 Lower Bounds on Exploration Time

The following three theorems are concerned with bounds on the amount of time required to explore graphs $\mathcal{G}$ that satisfy Assumption 2 for certain values of c. Throughout them, we consider only graphs $\mathcal{G}$ of order n with lifetime $L \geq c(n-1)$ in order to ensure that exploration is always possible.

Theorem 4. *There exists an infinite family of temporal graphs such that each member satisfies Assumption 2 for* $c = 3$ *(and thus also for all* $c \geq 3$*), has order* $n \in \{7, 10, 13, ...\}$*, and requires* $\Omega(n)$ *time steps to be explored in its entirety.*

Proof. Let $n = 3m + 1$ (for some $m \geq 2$) be the order of the temporal graph. Take an arbitrary $u \in V(\mathcal{G})$, then partition $V(\mathcal{G}) - \{u\}$ into three distinct parts $X = \{x_1, ..., x_m\}$, $Y = \{y_1, ..., y_m\}$ and $Z = \{z_1, ..., z_m\}$. In odd steps, we define the graph to consist of the components $X \cup \{u\}$ and $\{y_i, z_i\}$ for all $i \in [m]$. In even steps, it should consist of components $Y \cup \{u\}$ and $\{x_i, z_i\}$ for all $i \in [m]$. Furthermore, we (arbitrarily) set $s = x_1$. (An example of the construction can be seen in Fig. 1.) One can easily check that any pair of vertices in X (Y) can reach one another in at most 2 steps, and that vertices in X and Y are able to reach one another in at most 3 via u. Reaching any $x_j \in X$ ($y_j \in Y$) from any $z_i \in Z$ can also be achieved in at most three steps by waiting until the next step in which z_i and x_i (y_i) are contained in the same component, moving to x_i (y_i) in that step, and then to x_j (y_j) in the following step. Finally, consider reaching any vertex z_j from any vertex z_i ($i, j \in [m]$), starting at some time step t. Unless $i = j$, the quickest way to reach z_j from z_i is by first moving to the vertex $v \in \{x_i, y_i\}$ that lies in the same component as z_i in the current step. By construction, v will be contained in the same component as the vertex $v' \in \{x_j, y_j\}$; move to v' from v in step $t+1$, finally moving to z_j from v' during step $t+2$. In total this takes exactly 3 steps. Since any exploration schedule has to visit all $m = (n-1)/3$ z_i at some point and reaching one from the previous takes exactly 3 steps, it follows that any exploration schedule of $\mathcal{G}$ has duration $\Omega(n)$. To complete the proof, observe that any graph that satisfies Assumption 2 for a constant b also satisfies Assumption 2 for any $c > b$. $\square$

One direct consequence of Theorem 4 is that the aforementioned $c(n-1)$ upper bound on the length of exploration schedules in graphs satisfying Assumption 2 is in fact tight (asymptotically speaking) when $c \geq 3$. For the case in which $c = 2$, we now present a lower bound construction that requires $\Omega(\sqrt{n})$ steps to explore.

Theorem 5. *There exists an infinite family of graphs, the members of which satisfy Assumption 2 for* $c = 2$*, have order* $n \in \{4, 9, 16, ...\}$*, and require* $\Omega(\sqrt{n})$ *steps to be completely explored.*

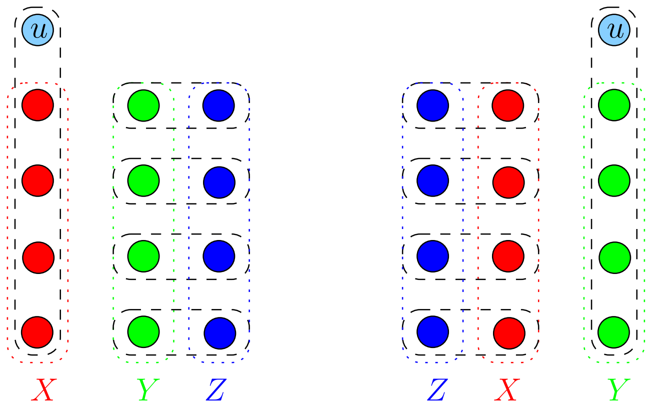

Fig. 1. The construction of Theorem 4 for $m = 4$. The left image is the graph in odd steps, with the graph in even steps displayed on the right. Black dashed lines mark the vertices contained in the components of either step.

Proof. Let $n = x^2$ for any $x \geq 2$. We now construct a graph $\mathcal{G}_n = \langle G_1, ..., G_L \rangle$, with $L = n$: Partition the vertex set into x parts of size x, arbitrarily labelling the vertices in the i-th part $v_{i,j}$ for $j \in [x]$. Arrange the vertices in the form of an x by x grid, with the first row consisting of the vertices $v_{1,1}, v_{1,2}, ..., v_{1,x}$, the second row of vertices $v_{2,1}$, $v_{2,2}, ..., v_{2,x}$, and so on and so forth (as in Fig. 2). We refer now to the collection of vertices $v_{1,j}, v_{2,j}, ..., v_{x,j}$ as the j-th column of the grid. Now, in every odd step $i = 1, 3, 5, ...$, let G_i be a partition of $V(\mathcal{G}_n)$ into the rows of the grid, and in every even step $i = 2, 4, 6, ...$, let G_i be a partition of $V(\mathcal{G}_n)$ into the columns. To see that $\mathcal{G}_n$ satisfies Assumption 2 (for $c = 2$), notice that in any pair of consecutive steps $t, t+1$ with $t \in [n-1]$, an agent can use one of those steps to change its row coordinate in the grid, and the other step to change its column coordinate.

To complete the proof, observe that in any step each component contains exactly $\sqrt{n}$ vertices. From this it follows that, during a single step, at most $\sqrt{n}$ unvisited vertices can be visited; hence at least $\Omega(\sqrt{n})$ steps are required of any exploration schedule. □

6.3 Upper Bounds on Exploration Time

The following result further concerns the case when $c = 2$, tightening the gap between the trivial $O(cn)$ upper bound and the $\Omega(\sqrt{n})$ lower bound of Theorem 5.

Theorem 6. *Any temporal graph $\mathcal{G}$ of order n satisfying Assumption 2 for $c = 2$ can always be explored in $O(\sqrt{n} \cdot \log n)$ time steps.*

Proof. We first show that for any consecutive pair of steps t and $t+1$, $\mathcal{G}$ consists of at most $\sqrt{n}$ components in at least one of these two steps. This is immediately obvious when each component of step t contains $\geq \sqrt{n}$ vertices (Case 1). Hence, we focus on the case in which at least one component of $\mathcal{G}$ during step t contains at most $\sqrt{n}$ vertices (Case 2). First, observe that in order for a graph to satisfy Assumption 2 for $c = 2$ it is required that, regardless of the time step t and the currently situated vertex/connected component, an agent can be positioned in any one of the step $t+1$ components of $\mathcal{G}$ by the start of that same step. This implies that the number of connected components of $\mathcal{G}$ in step $t+1$ is bounded from above by the size of the smallest component in step t (which by assumption of the case, is $\leq \sqrt{n}$).

Now, we show how to construct an exploration schedule W with duration $O(\sqrt{n} \cdot \log n)$. The idea is to divide the lifetime of $\mathcal{G}$ into consecutive blocks of three steps and within each, explore at least a $\frac{1}{\sqrt{n}}$ fraction of the currently unvisited vertices. More specifically, in the $(3j-2)$-th step ($j \geq 1$), apply the above case analysis, taking $i = 3j-1$ so that $i+1 = 3j$. If Case 1 applies, let $C^{\max}$ be the component at time step $3j-1$ which contains the largest number of previously unexplored vertices, resolving ties arbitrarily. Use time step $(3j-2)$ to move to some vertex that is contained in $C^{\max}$ in step $3j-1$ (by Assumption 2, this is always possible), exploring all unexplored vertices contained in $C^{\max}$ during step $3j-1$. If Case 2 applies, let $C^{\max}$ be the component at time step $3j$ which contains the largest number of previously unexplored vertices, again resolving ties arbitrarily. In this case, wait at the current vertex until time step $3j-1$, then move to some vertex contained in $C^{\max}$, which is again possible by Assumption 2, and explore all unexplored vertices contained in $C^{\max}$ during time step $3j$. In either case, the graph consists of at most $\sqrt{n}$ components during the time step in which the agent is positioned in $C^{\max}$. Let U be the set of previously unexplored vertices (which is initially the set $V - \{s\}$). The vertices in U are distributed amongst the $\leq \sqrt{n}$ components of step $3j-1$ or $3j$, and since $C^{\max}$ contains the largest number of them, it follows that $|C^{\max} \cap U| \geq \frac{|U|}{\sqrt{n}}$, as required.

Repeat the above process, exploring at least a $\frac{1}{\sqrt{n}}$ fraction of the previously unexplored vertices in each block of 3 consecutive steps until the number of unexplored vertices is less than 1. Since we began with $n-1$ unexplored vertices (we consider s to be automatically explored), we get that after $k = 3x$ steps (for any $x \in \{1, 2, 3, ...\}$), the remaining number of previously unvisited vertices is at most $n \cdot (1 - 1/\sqrt{n})^k$. We require that $n \cdot (1 - \frac{1}{\sqrt{n}})^k < 1$, which can be transformed into $n < (\frac{\sqrt{n}}{\sqrt{n}-1})^k$. Taking the logarithm of both sides yields

$$\log n < k \cdot \log \left(\frac{\sqrt{n}}{\sqrt{n}-1} \right) \iff k > \frac{\log n}{\log(1 + \frac{1}{\sqrt{n}-1})}.$$

Since $\log(1+x) > x/(1+x)$ for any $x > 0$, it then follows that the right-hand side of the previous inequality satisfies

$$\frac{\log n}{\log(1 + \frac{1}{\sqrt{n}-1})} < \frac{\log n}{\frac{1}{\sqrt{n}-1} \Big/ \frac{\sqrt{n}}{\sqrt{n}-1}} = \frac{\log n}{1/\sqrt{n}} = \sqrt{n} \cdot \log n.$$

Hence, as soon as the number of elapsed time steps k is greater than $\sqrt{n}\cdot\log n$, the number of remaining unexplored vertices is fewer than 1, and so the algorithm requires $O(\sqrt{n}\cdot\log n)$ steps to explore $\mathcal{G}$. □

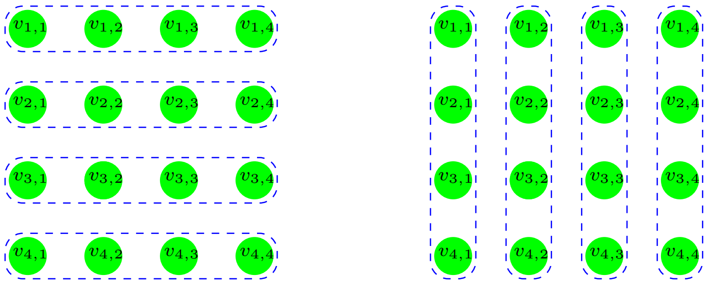

Fig. 2. The construction of Theorem 5 for $x = 4$. The left image is the graph in odd steps, with the graph in even steps displayed on the right. Blue dashed lines mark the vertices contained in the components of either step.

6.4 Inapproximability Results

The constructive nature of the proof of Theorem 6 implies the existence of a $\Theta(\sqrt{N}\cdot\log N)$-approximation algorithm for instances of Foremost-Non-Strict TEXP in which the given graph has order N and satisfies Assumption 2 for $c = 2$. The following result leaves only a $\Theta(\log N)$ gap between the best possible ratio achievable by any approximation algorithm for this problem and the ratio achieved by the algorithm implied by Theorem 6.

Theorem 7. *It is NP-hard to approximate solutions to instances of* Foremost-Non-Strict TEXP *that satisfy Assumption 2 for $c = 2$ with approximation guarantee $O(N^{\frac{1}{2}-\varepsilon})$ for any $\varepsilon > 0$, where N is the order of the given graph.*

Proof. Take an arbitrary instance I of 3SAT* and let $\mathcal{G}$ be the corresponding instance of Non-Strict TEXP generated via the construction of Theorem 3. Alter the construction so that there are now n^b $(b > 1)$ clause-copies corresponding to each of the m clauses of 3SAT* instance I, and call the resulting graph $\mathcal{G}'$. Furthermore, let the lifetime L of $\mathcal{G}'$ be ∞, and define the components of the resulting graph $\mathcal{G}'$ (of order $N = 2n + mn^b + 2(2n + mn^b)^2 = O(n^{2b+2})$) to be the same, during the first $n + 3$ steps, as the components of $\mathcal{G}$ (but with the additional clause-copy vertices added to the appropriate components). During all subsequent steps $t \in [n + 4, \infty]$, let the blue connectivity vertices alternate between being arranged as the rows and columns of a $2n+mn^b$ by $2n+mn^b$ grid, adding exactly one of the non-connectivity vertices (i.e., the literal/clause-copy vertices) to each of the components formed by the blue rows/columns. Take the red connectivity vertices and add them all to one arbitrary component in every subsequent step.

Next, observe that, by the same reasoning used in the proof of Theorem 3, $\mathcal{G}'$ admits an exploration schedule of length at most $n+3$ if and only if 3SAT* instance I is satisfiable. As a result, if $\mathcal{G}'$ cannot be fully explored by the end of the $(n+3)$-th step, then it must be that there exists one clause c in 3SAT* instance I whose corresponding clause-copy vertices have not yet all been explored. Note now that at most $n+3$ of these clause-copy vertices can have been explored during the first $n+3$ steps, and so at least $n^b-(n+3)$ remain unexplored. During any step $t \in [n+4,\infty]$, at most one of these remaining clause-copies can be explored, and so it follows that $\Theta(n^b)$ steps are required to explore them all. This implies that deciding whether $\mathcal{G}'$ can be explored in $\Theta(n)$ steps or whether $\Theta(n^b)$ steps are required also decides whether or not 3SAT* instance I is satisfiable. As such, it follows that approximating solutions to Non-Strict TEXP on graphs satisfying Assumption 2 for $c=2$ with approximation ratio strictly better than $\Theta(n^b)/\Theta(n) = \Theta(N^{\frac{1}{2}}/n^2) = \Theta(N^{\frac{1}{2}}/N^{\frac{1}{b+1}}) = \Theta(N^{\frac{1}{2}-\varepsilon'})$ is NP-hard, where $\varepsilon' = \frac{1}{b+1}$ and can be made arbitrarily close to 0 by choosing b large enough. The theorem follows for any $\varepsilon > 0$ by forcing $\varepsilon' \geq \varepsilon$ arbitrarily close to ε. $\square$

Theorem 8. *It is NP-hard to approximate solutions to instances of* Foremost-Non-Strict TEXP *by which Assumption 2 is satisfied for some $c \geq 3$ with approximation guarantee $O(N^{1-\varepsilon})$ for any $\varepsilon > 0$, where N is the order of the given graph.*

Proof. Let I be some instance of 3SAT* consisting of $n > 3$ variables v_i ($i \in [n]$) and $m = O(n)$ clauses. We wish to construct a non-strict temporal graph $\mathcal{G}$ (with lifetime $L = 3|V(\mathcal{G})|$) that satisfies Assumption 2 for $c=3$ but no $d<3$, and which admits an exploration schedule of length $O(n)$ if and only if I is satisfiable, otherwise requiring $\Omega(n^b)$ steps. To this end, we initially let $|V(\mathcal{G})| = N = 3mn^b + 1 = O(n^{b+1})$ for some $b \geq 2$. We take $2mn^b$ of the vertices in $V(\mathcal{G})$ and partition them into equisized sets $X = \{x_1, ..., x_{mn^b}\}$ and $Y = \{y_1, ..., y_{mn^b}\}$; let an additional vertex be known as u. The mn^b remaining vertices will be known as the *clause-copy* vertices $c_{j,k}$, with exactly n^b of them associated with j-th clause of I.

We now show how the components in each step of $\mathcal{G}$'s lifetime are to be arranged. In all steps $t \in [3n]$ such that $t \neq 3i$ for some $i \in [n]$, if t is odd, we place all $x \in X$ and u in the same connected component, whilst the mn^b vertices $y \in Y$ form a matching with the mn^b clause-copy vertices (this matching can be arbitrary, but will remain consistent in all considered steps). On the other hand, if $t \neq 3i$ and is even, then all $v \in Y \cup \{u\}$ form one component, whilst the vertices in X form a matching with the clause-copy vertices.

In all steps $t \in [3n]$ such that $t = 3i$ for some $i \in [n]$, if t is odd then all $v \in X \cup \{u\}$ form one connected component; let the start vertex $s = x_{mn^b}$. We create one component containing the vertex $y_1 \in Y$, along with all clause-copies corresponding to the clauses of I satisfied by setting $v_1 = 0$. To another component, we add the vertex $y_2 \in Y$, along with all clause-copies corresponding to the clauses of I satisfied by setting $v_1 = 1$. (In such steps, we will now refer to the components containing y_1 and y_2 as the 'true' and 'false' components, respectively.) All remaining clause-copies (i.e., those corresponding to clauses that are satisfied by neither a 0 nor 1 setting of v_i) will then form a matching with the remaining y_j ($j \in [mn^b] - \{1, 2\}$). (Note that there are always enough y_j to ensure this is possible, since at least one clause will be satisfied by either a 0 or 1 setting of each v_i, and so there can be at most $(m-1)n^b$ clause-copies to match with the $\leq mn^b - 2$ remaining y_j.) When t is even, the components are the same but the roles of sets X and Y are switched, so that now the components containing x_1 and x_2 are respectively the true and false components. During the steps $t \in [3n+1, 3N]$, the components alternate between being arranged as they are in odd/even steps $t' \in [3n]$ such that $t' \bmod 3 \neq 0$, depending on the parity of step t. It is straightforward to check that $\mathcal{G}$ satisfies Assumption 2 for $c = 3$. Moreover, consider any pair of clause-copies starting from any time step $t \geq 3n + 1$ and observe that 3 steps are in fact required to reach one from the other. We now demonstrate that I is satisfiable if and only if $\mathcal{G}$ is explorable in at most $3n$ steps, showing that at least $\Omega(n^b)$ steps are required otherwise.

($\Longrightarrow$) By arguments similar to those used in the proof of Theorem 1 we are able to construct from a satisfying assignment α for I an exploration schedule W of $\mathcal{G}$ with length at most $3n$. To do so, we use the steps $t \in [3i-2, 3i]$ to move to the true/false component in step $3i$ if α sets $v_i = 1/v_i = 0$, respectively.

($\Longleftarrow$) By arguments similar to those used in the proof of Theorem 1, we construct an assignment α for I by setting $v_i = 1$ if W visits the true component during the $3i$-th step, or set $v_i = 0$ if the false component is visited (with an arbitrary setting for v_i if W visits neither). This works since, if some clause of I had all n^b of its associated copies explored separately in steps when not in the true/false component of $\mathcal{G}$, then it would take at least $n^b > 3n$ (for $b \geq 2$ and $n > 3$) steps to visit them all, a contradiction to W's length being $\leq 3n$. As such, for every $j \in [m]$ there must be an $i \in [n]$ such that > 1 distinct copies associated with c_j, hence by construction all copies, are visited in the true/false component of step $3i$. It follows that α must be satisfying.

Moreover, if I has no satisfying assignment then any exploration schedule W must spend $\Theta(n^b)$ steps exploring all clause copies associated with ≥ 1 clause of I – otherwise there exists a schedule that visits all clause copies in a true/false component from which we could obtain a satisfying assignment for I. As a result, we may conclude that it is NP-hard to approximate instances of Non-Strict TEXP which satisfy Assumption 2 for any $c \geq 3$ with ratio strictly better than $\Theta(n^b)/3n = \Theta(n^{b-1}) = O(N^{\frac{b-1}{b+1}}) = O(N^{1-\frac{2}{b+1}}) = O(N^{1-\varepsilon'})$, where $\varepsilon' = 2/(b+1)$ can be made arbitrarily close to 0 by selecting b large enough. The theorem follows for any $\varepsilon > 0$ by forcing $\varepsilon' \geq \varepsilon$ arbitrarily close to ε. $\square$

7 Conclusion

We considered the problem of Non-Strict Temporal Exploration, a variant of the Temporal Exploration problem in which the requirement that edges in a candidate exploration schedule are crossed at strictly increasing timesteps is weakened, so that an edge may be crossed at a timestep greater than or equal to the timestep in which the last was crossed. We showed that deciding Non-Strict TEXP under these relaxed conditions is NP-complete.

The hardness of approximating solutions to Foremost-Non-Strict TEXP when the input graphs satisfy either of two distinct vertex-connectivity assumptions was also considered. For order n graphs satisfying the *pairwise vertex-togetherness* assumption (Assumption 1), we proved that it is NP-hard to approximate solutions with ratio $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$. For the second of these two assumptions, which posits that every pair of vertices can reach one another within $c = O(1)$ steps, we proved $O(n^{1-\varepsilon})$-inapproximability and $O(n^{\frac{1}{2}-\varepsilon})$-inapproximability in the $c \geq 3$ and $c = 2$ cases, respectively. Also shown was that, when $c = 2$, the graph of any yes-instance of Non-Strict TEXP can be explored in at most $O(\sqrt{n} \log n)$ timesteps. In complement to this, a lower bound construction which requires of any exploration algorithm at least $\Omega(\sqrt{n})$ steps was described. Closing the remaining $\Theta(\log n)$ gap presents an interesting direction for future work, as does the analysis of exploration time for graphs satisfying other assumptions that ensure exploration of a graph $\mathcal{G}$ is always possible. For example, one could examine the effect of some of the connectivity/reachability-ensuring measures presented in [18] within the non-strict model; [11] and [8] also consider temporal graphs with periodically-repeating properties whose effects could be interesting to explore within in the model considered here.

References

1. Akrida, E.C., Mertzios, G.B., Spirakis, P.G.: The temporal explorer who returns to the base. In: 11th International Conference on Algorithms and Complexity (CIAC 2019). LNCS, vol. 11485, pp. 13–24. Springer (2019). https://doi.org/10.1007/978-3-030-17402-6_2
2. Bodlaender, H.L., van der Zanden, T.C.: On exploring always-connected temporal graphs of small pathwidth. Information Processing Letters **142**, 68–71 (2019). https://doi.org/10.1016/j.ipl.2018.10.016
3. Brodén, B., Hammar, M., Nilsson, B.J.: Online and offline algorithms for the time-dependent TSP with time zones. Algorithmica **39**(4), 299–319 (2004). https://doi.org/10.1007/s00453-004-1088-z
4. 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). https://doi.org/10.1142/S0129054103001728
5. Casteigts, A., Flocchini, P., W., Q., Santoro, N.: Time-varying graphs and dynamic networks. IJPEDS **27**(5), 387–408 (2012)
6. Casteigts, A., Himmel, A.S., Molter, H., Zschoche, P.: The computational complexity of finding temporal paths under waiting time constraints. CoRR **abs/1909.06437** (2019), https://arxiv.org/abs/1909.06437
7. Di Luna, G.A., Dobrev, S., Flocchini, P., Santoro, N.: Live exploration of dynamic rings. In: 36th IEEE International Conference on Distributed Computing Systems (ICDCS 2016). pp. 570–579. IEEE (2016). https://doi.org/10.1109/ICDCS.2016.59

8. Erlebach, T., Hoffmann, M., Kammer, F.: On temporal graph exploration. In: 42nd International Colloquium on Automata, Languages, and Programming (ICALP 2015), Part I. LNCS, vol. 9134, pp. 444–455. Springer (2015). https://doi.org/10.1007/978-3-662-47672-7-36
9. Erlebach, T., Kammer, F., Luo, K., Sajenko, A., Spooner, J.T.: Two moves per time step make a difference. In: 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019). LIPIcs, vol. 132, pp. 141:1–141:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.141
10. Erlebach, T., Spooner, J.T.: Faster exploration of degree-bounded temporal graphs. In: 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018). LIPIcs, vol. 117, pp. 36:1–36:13. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2018). https://doi.org/10.4230/LIPIcs.MFCS.2018.36
11. Flocchini, P., Mans, B., Santoro, N.: On the exploration of time-varying networks. Theoretical Computer Science **469**, 53–68 (2013). https://doi.org/10.1016/j.tcs.2012.10.029
12. Fluschnik, T., Molter, H., Niedermeier, R., Renken, M., Zschoche, P.: Temporal graph classes: A view through temporal separators. Theor. Comput. Sci. **806**, 197–218 (2020). https://doi.org/10.1016/j.tcs.2019.03.031
13. Gotoh, T., Flocchini, P., Masuzawa, T., Santoro, N.: Tight bounds on distributed exploration of temporal graphs. In: 23rd International Conference on Principles of Distributed Systems (OPODIS 2019). LIPIcs, vol. 153, pp. 22:1–22:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.OPODIS.2019.22
14. Ilcinkas, D., Klasing, R., Wade, A.M.: Exploration of constantly connected dynamic graphs based on cactuses. In: 21st International Coloquium on Structural Information and Communication Complexity (SIROCCO 2014). LNCS, vol. 8576, pp. 250–262. Springer (2014). https://doi.org/10.1007/978-3-319-09620-9_20
15. Ilcinkas, D., Wade, A.M.: Exploration of the T-interval-connected dynamic graphs: The case of the ring. In: 20th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2013). LNCS, vol. 8179, pp. 13–23. Springer (2013). https://doi.org/10.1007/978-3-319-03578-9_2
16. Kempe, D., Kleinberg, J.M., Kumar, A.: Connectivity and inference problems for temporal networks. J. Comput. Syst. Sci. **64**(4), 820–842 (2002). https://doi.org/10.1006/jcss.2002.1829
17. Michail, O.: An introduction to temporal graphs: An algorithmic perspective. Internet Mathematics **12**(4), 239–280 (2016). https://doi.org/10.1080/15427951.2016.1177801
18. Michail, O., Chatzigiannakis, I., Spirakis, P.G.: Causality, influence, and computation in possibly disconnected synchronous dynamic networks. Journal of Parallel and Distributed Computing **74**(1), 2016–2026 (2014). https://doi.org/10.1016/j.jpdc.2013.07.007
19. Michail, O., Spirakis, P.G.: Traveling salesman problems in temporal graphs. Theor. Comput. Sci. **634**, 1–23 (2016). https://doi.org/10.1016/j.tcs.2016.04.006
20. Sobin, C.C., Raychoudhury, V., Marfia, G., Singla, A.: A survey of routing and data dissemination in delay tolerant networks. J. Netw. Comput. Appl. **67**, 128–146 (2016). https://doi.org/10.1016/j.jnca.2016.01.002
21. Tovey, C.A.: A simplified NP-complete satisfiability problem. Discrete Applied Mathematics **8**(1), 85–89 (1984). https://doi.org/10.1016/0166-218X(84)90081-7