Exploiting Automorphisms of Temporal Graphs for Fast Exploration and Rendezvous

Konstantinos Dogeas
Department of Computer Science, Durham University, UK

Thomas Erlebach
Department of Computer Science, Durham University, UK

Frank Kammer
THM, University of Applied Sciences Mittelhessen, Gießen, Germany

Johannes Meintrup[1]
HM, University of Applied Sciences Mittelhessen, Gießen, Germany

William K. Moses Jr.
Department of Computer Science, Durham University, UK

Abstract

Temporal graphs are dynamic graphs where the edge set can change in each time step, while the vertex set stays the same. Exploration of temporal graphs whose snapshot in each time step is a connected graph, called connected temporal graphs, has been widely studied. In this paper, we extend the concept of graph automorphisms from static graphs to temporal graphs and show for the first time that symmetries enable faster exploration: We prove that a connected temporal graph with n vertices and orbit number r (i.e., r is the number of automorphism orbits) can be explored in $O(rn^{1+\epsilon})$ time steps, for any fixed $\epsilon > 0$. For $r = O(n^c)$ for constant $c < 1$, this is a significant improvement over the known tight worst-case bound of $\Theta(n^2)$ time steps for arbitrary connected temporal graphs. We also give two lower bounds for temporal exploration, showing that $\Omega(n \log n)$ time steps are required for some inputs with $r = O(1)$ and that $\Omega(rn)$ time steps are required for some inputs for any r with $1 \leq r \leq n$.

Moreover, we show that the techniques we develop for fast exploration can be used to derive the following result for rendezvous: Two agents with different programs and without communication ability are placed by an adversary at arbitrary vertices and given full information about the connected temporal graph, except that they do not have consistent vertex labels. Then the two agents can meet at a common vertex after $O(n^{1+\epsilon})$ time steps, for any constant $\epsilon > 0$. For some connected temporal graphs with the orbit number being a constant, we also present a complementary lower bound of $\Omega(n \log n)$ time steps.

2012 ACM Subject Classification Theory of computation → Dynamic graph algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases dynamic graphs, parameterized algorithms, algorithmic graph theory, graph automorphism, orbit number

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.55

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2312.07140` [25]

Funding *Johannes Meintrup*: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 379157101.

[1] Corresponding Author

1 Introduction

For many decades, graph theory has been a tool used to model and study many real world problems and phenomena [15]. A usual assumption for many of these problems is that the graphs have a fixed structure. However, there are quite a number of cases where the structure of a system changes over time. For example, consider the problem of routing in transportation networks (roads, rails) where specific connections can become unavailable (due to a disaster) or they are active during specific times (due to safety). Such scenarios can be modeled using temporal graphs, a sequence of graphs over the same vertex set where the edges possibly change in each time step. The temporal graph setting has received significant interest from the research community in the recent past as seen in recent surveys [19, 48].

In this paper, we study two problems on temporal graphs. The first is the *temporal exploration problem* (TEXP), which has been studied, e.g., by Michail and Spirakis [49, 50] and by Ilcinkas et al. [37]. It requires an agent to explore all vertices of the temporal graph as quickly as possible. The second is the *temporal rendezvous problem* (TRP), which we formulate for the first time in this paper. It requires two heterogeneous agents (in terms of the programs they run) to rendezvous on a temporal graph when they cannot communicate with one another. For both problems, we assume that each agent has complete knowledge of the temporal graph in advance (a common assumption [3, 14, 27, 28, 30, 31, 37, 49, 50, 62]). However, in the case of TRP the agents may have different names for the same vertices, i.e., the local labels of the vertices may be different.

The problem of exploration has been well studied in the static setting since it was introduced in 1951 by Shannon [59]. It has also been intensively studied in the temporal graph setting since 2014 (see all references from the previous paragraph). On an application-oriented note, the problem captures the setting where a person is trying to visit various parts of a city using public transportation. For example, train schedules involve multiple train stations (vertices) with trains running between them at different times (i.e., repeatedly changing edges). Thus, planning a visit to multiple destinations over a given day using railways is an example of solving TEXP.

The rendezvous problem can be broadly categorized into two types: symmetric and asymmetric rendezvous. The version where agents have the same strategy (symmetric rendezvous) was introduced by Alpern [10]. The version where agents can have distinct strategies (asymmetric rendezvous) was introduced by Alpern [7] and is the focus of research in this paper. TRP is a natural extension of the asymmetric rendezvous problem to the dynamic setting. As a real world example, consider a pair of tourists who want to explore a city together and have to agree on a strategy to meet up in case they are separated and their cell phones die. In this scenario, they may use public transportation (dynamically changing network) to meet and agree in advance to use different strategies that guarantee that they meet quickly.

In this paper, we present results that extend the literature in two ways. Firstly, we formalize the TRP problem in the setting where agents have complete knowledge of the temporal graph a priori and develop good upper and lower bounds for it. Secondly, we utilize interesting structural properties (namely automorphisms of a graph and associated orbits of vertices) in order to analyze bounds on the number of time steps required by temporal walks that solve certain problems. In particular, we develop upper and lower bounds for both TEXP and TRP that leverage the aforementioned graph properties. To the best of our knowledge, this is the first work that takes advantage of these graph properties to study problems in temporal graphs.

1.1 Our Contributions

We present results for two problems: first, the TEMPORAL EXPLORATION PROBLEM (TEXP) which, for a given temporal graph $\mathcal{G}$, asks for a temporal walk that visits all vertices of $\mathcal{G}$, and secondly, the TEMPORAL RENDEZVOUS PROBLEM (TRP), which considers two agents that try to meet in the given temporal graph, meaning they must be stationed at the same vertex in the same time step.

One of our primary contributions is formalizing the problem of TRP in Section 2, and in doing so extending the problem of asymmetric rendezvous to the temporal graph setting where agents have complete knowledge of the temporal graph in advance. Another significant contribution is that we show how to leverage the use of a structural graph property, namely the automorphism group of a temporal graph and the associated notion of orbits, to bound the number of time steps required by algorithms we devise for these problems. To the best of our knowledge, this work is the first instance of leveraging such properties to study any problem on temporal graphs apart from a practical implementation of a generator for listing all non-isosomorphic simple temporal graphs [18]. Intuitively, an automorphism of a graph is a mapping from the set of vertices of the graph to the same set of vertices that preserves the neighborhood relation between the vertices. The set of automorphisms of a temporal graph consists of the *intersection* of the sets of automorphisms of the graph at each time step. The set of automorphisms of the temporal graph, along with function composition as group operation, forms the automorphism group of the temporal graph. An orbit of the automorphism group of a temporal graph is a maximal set of vertices such that each vertex can be mapped to any other vertex in the set via an automorphism of the group. Intuitively, the vertices in an orbit look indistinguishable to an agent that has full information about the temporal graph but without meaningful vertex labels. As the agents are able to compute a consistent numbering of the orbits (Lemma 13), they can agree to meet in some specific orbit, but not at a specific vertex. Therefore, temporal graphs where all orbits are large (or that even have a single orbit containing all vertices) appear to be the most challenging graphs for solving TRP. Our result providing fast exploration schedules in temporal graphs with few orbits is therefore a crucial ingredient for enabling our solution to TRP to handle all possible temporal graphs (including those with a single orbit).

We give precise definitions and present further preliminaries in Section 2. Then, we introduce some useful utilities related to automorphisms in Section 3. These will be used in later sections, where we present the following results.

Upper bounds. In Section 4, we develop a deterministic algorithm to solve TEXP in $O(rn^{1+\epsilon})$ time steps for any fixed $\epsilon > 0$ (see Corollary 12), where n is the number of vertices in the temporal graph and r is the number of orbits of the automorphism group of the temporal graph. Note that r can range in value from 1 to n. Thus, for $r = O(n^c)$ for constant $c < 1$, this is a significant improvement over the known tight worst-case bound of $\Theta(n^2)$ time steps for arbitrary connected temporal graphs [27]. In Section 5, we leverage this algorithm for TEXP to develop a deterministic solution for TRP using $O(n^{1+\epsilon})$ time steps for any fixed $\epsilon > 0$ (see Theorem 14). Our focus is on bounding the time steps of the temporal walks required to solve TEXP and TRP, and not on optimizing the running time of the respective algorithms to compute such walks.

Lower bounds. We complement our algorithms with lower bounds for both TEXP and TRP in Section 6. In particular, we design an instance of TRP such that any solution for it requires $\Omega(n \log n)$ time steps (see Theorem 16). We then show how this translates to a lower bound of $\Omega(n \log n)$ time steps for some instances of TEXP where the temporal graph at hand has an orbit number $r = O(1)$ (see Corollary 17). By revisiting the lower bound of [27] for TEXP, which focused on arbitrary temporal graphs, and studying it through the lens of automorphisms and orbits, we can obtain a more fine-grained lower bound of $\Omega(rn)$ time steps (see Lemma 15) for some temporal graphs with orbit number $1 \leq r \leq n$. Notice that the multiplicative gap between our upper and lower bounds for both problems is only a factor of $O(n^\epsilon)$.

Relevance of the orbit number. It is well known in graph theory that almost all (static) graphs are *rigid* [12], meaning that the automorphism group contains only the trivial identity function. This is in contrast to observations in practice. Many real-world graphs have non-trivial automorphisms. A recent analysis of real-world graphs in the popular database `https://networkrepository.com` showed that over 70% of the analyzed graphs had non-trivial automorphisms [12]. One may reasonably expect that real-world temporal graphs have similar properties. Symmetries are also abundant in graphs arising in chemistry [11], which have been studied in temporal settings as well [57]. We believe that these observations provide a strong motivation for studying temporal graph problems for temporal graphs with fewer than n orbits, e.g., with algorithms parameterized via the number of orbits. Furthermore, we emphasize that our results for TRP hold for all temporal graphs, independent of the orbit number parameter, even though they utilize techniques we develop for TEXP parameterized by the orbit number. Roughly speaking, for orbit number r, the penalty factor r in the number of time steps to solve TEXP is saved when solving TRP by focusing only on a smallest orbit of size at most n/r.

1.2 Technical Overview and Challenges

Firstly, we give an intuitive overview of our upper bound results. The concepts used in this section are more precisely defined in the next sections.

For TEXP, we first consider the problem of visiting all the vertices of one orbit S. One key insight is that, if we have a temporal walk W that visits k vertices of S, we can use the automorphisms of the temporal graph to transform W into other walks that visit different sets of k vertices of S (Lemma 3). Therefore, even if the k vertices visited by W have already been explored earlier, we can transform W into a temporal walk W' that visits a "good" number of previously unexplored vertices of S. The number of previously unexplored vertices of S that W' is guaranteed to visit increases with the number of possible start vertices in S that we allow for W', but the larger that set X of possible start vertices is, the longer it may take the agent to reach the best start vertex in that set. A challenge is to analyze this tradeoff. By carefully relating the size of X to the guaranteed number of unexplored vertices that a walk W' starting at a vertex in X can visit (Corollary 4), we manage to balance the number of time steps needed to move to the start vertex of W' and the number of previously unexplored vertices that W' visits. To show that vertices of X can be reached quickly, we study the structure of edges that connect vertices in different orbits (Lemma 5) and use it to analyze reachability between orbits (Lemmas 6 and 7).

We then employ a recursive construction: We recursively construct a temporal walk W_1 that visits k vertices of S, concatenate W_1 with a temporal walk that moves quickly from the endpoint of W_1 to a good start vertex in S for what follows, and then use a second recursively constructed walk (transformed via an automorphism to a "good" temporal walk W_2 starting in a vertex of S) to visit "nearly" k further vertices of S. A careful analysis then shows that in this way we can visit a constant fraction of the vertices of S in $O(r|S|^{1+\epsilon})$ steps

(Lemma 9 and Corollary 10). We then show that the concatenation of $O(\log |S|)$ such walks (each one again transformed via an automorphism to maximize the number of newly explored vertices) suffices to visit all vertices of S in $O(r|S|^{1+\epsilon} + n \log |S|)$ time steps (Theorem 11), which can be bounded by $O(rn^{1+\epsilon'})$ time steps). By visiting the r orbits one after another, we can finally show that the whole temporal graph can be explored in $O(rn^{1+\epsilon'})$ time steps (Corollary 12).

For TRP, a simple and fast solution one first thinks of is to have the agents simply meet at a vertex with a specific label. However, this is not feasible in the model we consider. In particular, we assume that the agents cannot communicate and, while both agents have complete information about the temporal graph, they do not have access to consistent vertex labels. As such, the agents are unable to agree upon the same vertex based on the vertex label. We rely, instead, on structural graph properties and let the agents meet at a vertex in a smallest orbit: One agent moves to any vertex in that orbit, and the other searches all vertices in that orbit. The first challenge is for the agents to independently identify the same orbit in which to meet. We show that this can be done by letting each agent enumerate all temporal graphs, together with colorings of their orbits, until it encounters for the first time a temporal graph that is isomorphic to the input graph in which TRP is to be solved. In this way both agents obtain the same colored temporal graph and can independently select, among all orbits of smallest size, the one with smallest color (Lemma 13). Then one agent moves to a vertex in that orbit S, while the other searches all vertices of S. The challenge here is to deal with orbits that are large, because previously known techniques would require $\Theta(n^2)$ time steps to explore an orbit of size $\Omega(n)$. Fortunately, this is where our above-mentioned results for exploration of (orbits of) temporal graphs come to the rescue: As $|S| \leq n/r$, we have $O(r|S|^{1+\epsilon} + n \log |S|) = O(n^{1+\epsilon'})$, and hence S can be explored (and TRP solved) in $O(n^{1+\epsilon'})$ time steps (Theorem 14).

Now we turn to our lower bounds. For TEXP, we first observe that the existing lower bound construction that shows that $\Omega(n^2)$ time steps are necessary for exploration on some temporal graphs uses temporal graphs with $\frac{n}{2}+1$ orbits. By slightly varying the construction, we can show for any r in the range from 1 to n that there are temporal graphs with r orbits that require $\Omega(rn)$ time steps for exploration (Lemma 15). Our main lower bound result is the lower bound of $\Omega(n \log n)$ for TRP in temporal graphs with a single orbit (Theorem 16). The temporal graph is a cycle on $n = 2^m - 1$ vertices in every step, and the edges change every $\lfloor n/16 \rfloor$ time steps. Each period of $\lfloor n/16 \rfloor$ time steps in which the edges do not change is called a *phase*. We number the vertices of the cycle from 0 to $n-1$ and consider the binary representation of the vertex labels. In phase 1, each vertex j is adjacent to $j+1$ and $j-1$, while in any phase $i > 1$, it is adjacent to $j + 2^{2i}$ and $j - 2^{2i}$ (with all computations done modulo n). As all vertices are indistinguishable to the agents, we can force each agent to make the same movements no matter where we place it initially. By fixing the five highest-order bits of the agents' start positions in a certain way, we can ensure that the agents do not meet in the first phase, no matter how the remaining bits of their start positions are chosen. Depending on the positions where the agents end up at the end of each phase (relative to their start position in that phase), we fix a certain number of bits of the binary representations of the start positions of the agents: After the first phase, we fix the lowest four bits, and after each further phase, we fix the next-higher two bits. We can show that this is sufficient to ensure that the agents start the next phase at vertices that are sufficiently far from each other in the cycle of that phase. This process can be repeated $\Omega(\log n)$ times, showing that it takes $\Omega(n \log n)$ time steps for the agents to meet. This lower bound for TRP implies also that exploration of the constructed temporal graph requires $\Omega(n \log n)$ steps (Corollary 17).

1.3 Related Work

There is a divide in the research on temporal graphs with respect to the amount of knowledge known in advance by the agents on the graph. When complete knowledge of the temporal graph is known in advance to the agents, as is assumed in this paper, the setting is sometimes referred to as the *post-mortem* setting (see, e.g., Santoro [58]). Since we consider scenarios where agents plan temporal walks using advance knowledge of future time steps, we refer to the post-mortem setting as the *clairvoyant* setting in the remainder of this section.

The clairvoyant setting is in contrast to the *live* setting where agents only have partial knowledge of the temporal graph, and the solutions to problems in these different settings are of a different nature. The reader can refer to the survey by Di Luna [22] for more information on problems and solutions in the live setting. We restrict the rest of this related work section to work in the clairvoyant setting.

Exploration. We trace the progress of solving instances of TEXP, first by identifying which assumptions are required for the problem to even be solvable, and then by identifying properties that were leveraged to give faster and faster solutions.

Michail and Spirakis [49, 50] showed that it is **NP**-complete to decide whether a given temporal graph with a given *lifetime*, i.e., the number of time steps it exists, is explorable when no assumptions are made on the input graph. This holds even if the graph is connected in every time step, termed *connected* (and sometimes called *always-connected* in the literature). Even when a restriction is placed on the *underlying graph*, i.e., the union of the graphs at each time step, such that the underlying graph has pathwidth at most 2, Bodlaender and van der Zanden [14] showed that TEXP is **NP**-complete.

However, the exploration is always possible when the lifetime of the graph is sufficiently large. In particular, Michail and Spirakis [50] showed that a connected graph with n vertices may be explored in $O(n^2)$ time steps. For the rest of the related work on TEXP, we focus on the case of connected temporal graphs with a sufficiently large lifetime. Erlebach et al. [27] showed that exploration on arbitrary temporal graphs takes $\Omega(n^2)$ time steps. By restricting their study of temporal graphs to those where the underlying graph belongs to a special graph class, however, they showed that TEXP can be solved in $o(n^2)$ time steps in several such cases. In particular, when the underlying graph is planar, has bounded treewidth k, or is a $2 \times n$ grid, they showed that TEXP can be solved in $O(n^{1.8} \log n)$ time steps, $O(n^{1.5} k^{1.5} \log n)$ time steps, and $O(n \log^3 n)$ time steps, respectively. They also showed a lower bound of $\Omega(n \log n)$ when the underlying graph is a planar graph of degree at most 4.

The study of TEXP when the temporal graph is restricted continued in several papers. Taghian Alamouti [62] showed that TEXP can be solved in $O(k^2 (k!) (2e)^k n)$ time steps when the underlying graph is a cycle with k chords. Adamson et al. [3] improved this to $O(kn)$ time steps. They also improved the upper bounds on TEXP for underlying graphs that have bounded treewidth k or are planar to $O(kn^{1.5} \log n)$ and $O(n^{1.75} \log n)$, respectively. In addition, they strengthened the lower bound for underlying planar graphs by showing that even if the degree is at most 3, the lower bound is $\Omega(n \log n)$. Erlebach et al. [28] further improved work on bounded degree underlying graphs by showing that TEXP can be solved in $O(n^{1.75})$ time steps for such temporal graphs. Ilcinkas et al. [37] showed that when the underlying graph is a cactus, the exploration time is $2^{\Theta(\sqrt{\log n})} n$ time steps.

Other variants of TEXP have been studied where the problem is slightly different, or the edges of the temporal graph vary in some particular way (e.g., periodically, T-interval connected, k-edge deficient), or multiple agents explore the graph [1, 2, 5, 17, 30, 31, 38].

Rendezvous. Symmetric rendezvous [10] and asymmetric rendezvous [7] have received much interest over the years, resulting in numerous surveys [8, 9, 32, 40, 41, 54, 55] being written on the problems covering different settings. In this work, we are the first to extend asymmetric rendezvous to temporal graphs in the clairvoyant setting. Note that there has been previous work [16, 21, 23, 51, 53, 60, 61] that has studied multi-agent rendezvous (also called gathering) in a dynamic graph setting, but that was in the live setting. For an overview of rendezvous in static graphs we refer the reader to [9, 41]. Rendezvous has been studied in deterministic settings [13, 56] and asynchronous settings [24, 43], and has been shown to be solvable in logarithmic space [20].

Other Related Work. Other problems have also been studied in the temporal graph setting, e.g., matchings [46], separators [34, 63], vertex covers [6, 36], containments of epidemics [26], Eulerian tours [17, 44], graph coloring [45, 47], network flow [4], treewidth [33] and cops and robbers [29, 52]. See the survey by Michail [48] for more information.

2 Preliminaries

Basic terminology. We use standard graph terminology. All static graphs are assumed to be simple and undirected. We write $[x] = \{1, \ldots, x\}$ for any integer x.

A *temporal graph* $\mathcal{G} = (G_1, \ldots, G_\ell)$ is a sequence of static graphs, all with the same vertex set V. For a temporal graph $\mathcal{G}$, the graph $G = (V, \bigcup_{t\in[\ell]} E(G_t))$ is the *underlying graph of* $\mathcal{G}$. We call $\mathcal{G}$ connected if each G_t for $t \in [\ell]$ is connected. We use n to refer to the number of vertices of the (temporal) graph under consideration. A *temporal walk* W is a walk in a temporal graph $\mathcal{G}$ that is time respecting, i.e., traverses edges in strictly increasing time steps. For a temporal walk W starting at time step t we write $W = (u_1, u_2, u_3, \ldots)$ to mean the temporal walk starts at vertex u_1 in time step t, is at vertex u_2 at the beginning of time step $t + 1$, and so forth. Note that subsequent vertices in the temporal walk can be the same vertex, i.e., we can wait for an arbitrary number of time steps at a vertex. We say that a temporal walk *spans* x time steps if it starts at some time step t and ends at some time step t' with $t' - t = x$. We assume that the lifetime ℓ is large enough such that a desired temporal walk can be constructed. For all our results it is enough to assume $\ell \geq n^2$, as n^2 time steps suffice for TEXP [27] and therefore also for TRP. For any function $f : X \to X$ for a universe X we write $f^i(x)$ when we mean applying the function i-times iteratively for any $x \in X$ and integer i, e.g., $f^2(x) = f(f(x))$. We use $\circ$ to denote function composition, i.e., for two functions $f : X \to X$ and $g : X \to X$, we denote with $f \circ g$ the function that maps $x \in X$ to $f(g(x))$.

Isomorphisms and automorphisms. Two static graphs G and H are *isomorphic* exactly if a bijection $\theta : V(G) \to V(H)$ (called an *isomorphism*) exists with the following property: two vertices u, v are adjacent in G exactly if $\theta(u)$ is adjacent to $\theta(v)$ in H. We write $G \cong H$ if G is isomorphic to H. An *automorphism* is an isomorphism from a graph G to itself. The set of all automorphisms of a graph G forms a group $\mathrm{Aut}(G)$, with $\circ$ as group operation. Refer to [35, 39, 42] for further reading on the topic of isomorphisms and automorphisms of graphs.

For a temporal graph $\mathcal{G}$ with lifetime ℓ we denote with $\mathrm{Aut}(\mathcal{G})$ the set of all functions σ such that σ is an automorphism of each graph G_t at every time step $t \in [\ell]$ (and hence also an automorphism of the underlying graph G). $\mathrm{Aut}(\mathcal{G})$ with $\circ$ as group operation is the automorphism group of the temporal graph $\mathcal{G}$. We remark that the automorphism group of the temporal graph $\mathcal{G}$ is the intersection of the automorphism groups of the graphs G_t for

$t \in [\ell]$ and that the automorphism group of $\mathcal{G}$ can be viewed as the automorphism group of the underlying graph G of $\mathcal{G}$ with edge labels such that each edge is labeled with the set of time steps in which it occurs.

The *orbit* of a vertex u in a temporal graph $\mathcal{G}$ with respect to $\mathrm{Aut}(\mathcal{G})$ is the set $V' \subseteq V$ of all vertices that u can be mapped to by automorphisms in $\mathrm{Aut}(\mathcal{G})$. Note that, if V' is the orbit of u, then the orbit of every vertex in V' is also V': For any two vertices v, v' in V', the automorphisms σ and σ' that map u to v and v', respectively, can be composed to the automorphism $\sigma' \circ \sigma^{-1}$ that maps v to v'. Furthermore, there cannot be an automorphism ρ that maps v to a vertex outside V' because $\rho \circ \sigma$ would then be an automorphism that maps u to a vertex outside V'; a contradiction to the definition of the orbit V' of u. We denote by $\mathcal{G}/\mathrm{Aut}(\mathcal{G})$ the set of all orbits of the vertices of $\mathcal{G}$. Note that this set forms a partition of V. We call $|\mathcal{G}/\mathrm{Aut}(\mathcal{G})|$ the *orbit number* of $\mathcal{G}$. We call an edge $\{u, v\} \in E(G_t)$ for $t \in [\ell]$ an *orbit boundary edge* if u and v are in different orbits, and *inner orbit edge* otherwise. Two orbits connected by an orbit boundary edge in time step t are called *adjacent (in time step t)*.

We use automorphisms to transform temporal walks, as outlined in the following. For a temporal graph $\mathcal{G}$ with lifetime ℓ and any automorphism $\sigma \in \mathrm{Aut}(\mathcal{G})$ and any temporal walk $W = (u_t, u_{t+1}, \ldots, u_{t'})$ that starts at time t and ends at time t' with $t, t' \in [\ell]$, we say we *apply σ to W* when we construct the temporal walk $W' = (\sigma(u_t), \sigma(u_{t+1}), \ldots, \sigma(u_{t'}))$.

Temporal Exploration Problem. The TEMPORAL EXPLORATION PROBLEM (TEXP) is defined for a given temporal graph $\mathcal{G}$ with vertex set V and lifetime ℓ and asks for the existence of a temporal walk W such that W starts at a given vertex $u \in V$ and visits all vertices of V. In connected temporal graphs with sufficiently large lifetime such a walk always exists, and this is the setting we consider throughout this work. As such, the question of existence is no longer of interest, and instead we focus on the time span of temporal walks that start at u at time step 1 and visit all (or a certain set of) vertices.

The following is an adaptation of a result of Erlebach et al. [27], slightly modified to fit our notation and use case. Intuitively speaking, the lemma states that with every extra time step at least one additional vertex becomes reachable. Thus, starting from any vertex at any time step, we can reach any particular other vertex in $n-1$ steps. Consequently, there is always a temporal walk that visits all vertices of $\mathcal{G}$ within n^2 time steps.

▶ **Lemma 1** (Reachability, [27]). *Let $\mathcal{G}$ be a connected temporal graph with vertex set V and lifetime ℓ. Denote by $R_{t,t'}(u)$ the set of vertices reachable by some temporal walk starting at vertex $u \in V$ at time step $t \in [\ell]$ and ending at time step $t' \in [\ell]$ with $t' \geq t$. If $R_{t,t'}(u) \neq V$ and $t' < \ell$, then $R_{t,t'}(u) \subsetneq R_{t,t'+1}(u)$.*

Temporal Rendezvous Problem. We consider a problem we call TEMPORAL RENDEZVOUS PROBLEM (TRP) defined as follows. Let $\mathcal{G}$ be a temporal graph with vertex set V and lifetime ℓ. Two agents a_1, a_2 are placed at arbitrary vertices $u_1, u_2 \in V$ (respectively) in G_1 of $\mathcal{G}$. They compute respective temporal walks W_1 and W_2 such that the agents meet at the same vertex in the same time step at least once during these walks. The agents have the full information of $\mathcal{G}$ available, but they cannot communicate with each other. Furthermore, they do not know the location of the other agent, and the vertex labels that one agent sees can be arbitrarily different from the vertex labels that the other agent sees. The lack of consistent labels prohibits trivial solutions such as the two agents agreeing to meet at a vertex with a specific label, e.g., the lowest labeled vertex. We call such agents *label-oblivious*. A solution to TRP is a pair of possibly different programs p_1 and p_2 for agents a_1 and a_2, respectively, that the agents use to compute and execute their temporal walks. The respective start positions $u_1, u_2 \in V$ of the agents (and the vertex labels that each agent sees) are chosen by an all-knowing adversary once a solution (p_1, p_2) is provided. As we assume that $\ell \geq n^2$, there is always a solution that ensures that the agents meet: Agent a_1 simply waits at its start vertex, while agent a_2 explores the whole graph and visits every vertex, which is always possible within n^2 time steps as mentioned above. Therefore, we are interested in solutions that enable the agents to meet as early as possible and aim to obtain worst-case bounds significantly better than n^2 on the number of steps that are required for TRP.

3 Automorphism Utilities

We now introduce some helpful utilities regarding automorphisms that we use in the following sections. Intuitively, we build up to a framework that allows us to transform a temporal walk W into a temporal walk W' that visits more vertices that are desirable (with respect to the exploration goal) than W does, by applying a well-chosen automorphism to W. The following is needed to give specific guarantees that the transformed temporal walks must uphold. Throughout this section we use and extend techniques of the field of algebraic graph theory [35, 39, 42], adapted to our specific use cases.

For this, we begin with some definitions. Let $\mathcal{G}$ be a temporal graph with vertex set V and let $S \in \mathcal{G}/\mathrm{Aut}(\mathcal{G})$ be any orbit. For any $X \subseteq S$ and any $u \in S$ denote with $\mathrm{Aut}(\mathcal{G})[u, X]$ the set of all automorphisms $\sigma \in \mathrm{Aut}(\mathcal{G})$ that map u to any vertex $v \in X$. We use the shorthand $\mathrm{Aut}(\mathcal{G})[u, x]$ for $\mathrm{Aut}(\mathcal{G})[u, \{x\}]$. We can then show the following.

▶ **Lemma 2.** *Let $S \in \mathcal{G}/Aut(\mathcal{G})$ be any orbit. Then $|Aut(\mathcal{G})[u, x_1]| = |Aut(\mathcal{G})[u, x_2]|$ for any $u, x_1, x_2 \in S$.*

Let $S \in \mathcal{G}/\mathrm{Aut}(\mathcal{G})$. We now consider a special 2-dimensional *automorphism matrix* $\mathcal{M}_{u,Y,X}$ for any $u \in S$, $Y \subseteq S$, and $X \subseteq S$. It has columns $C_1, C_2, \ldots, C_{|Y|+1}$, and a row for each $\sigma \in \mathrm{Aut}(\mathcal{G})[u, X]$. We refer to the row for some σ simply as row σ. The entry in row σ of column C_1 is $\sigma(u)$. Each vertex $y \in Y$ is assigned a unique column among $C_2, \ldots C_{|Y|+1}$ in an arbitrary way. The entry in row σ of the column to which y is assigned is $\sigma(y)$. We now give some intuition about the application of an automorphism matrix. Assume that we are constructing a temporal walk W that has already visited a set $T \subsetneq S$ and we want to extend it to visit the vertices of $S \setminus T$ (with S an arbitrary orbit). To facilitate this extension we first construct a temporal walk W' that visits at least a certain number of vertices of S, but that is not guaranteed to visit vertices of $S \setminus T$. This walk W' cannot be used to extend W in a meaningful way. Instead, using the automorphism matrix we show that there always exists an automorphism $\sigma \in \mathrm{Aut}(\mathcal{G})$ that we can apply to W' to obtain a temporal walk W_σ that visits a guaranteed fraction of vertices of $S \setminus T$. We can then use W_σ as our desired extension. Figure 2 visualizes this concept. In later sections we will show how repeated application of such extensions leads to temporal walks that visit all vertices of an orbit S. Note that the value p in the following lemma represents the fraction of vertices that we still need to visit compared to all vertices in the orbit under consideration. While T is a set of already visited vertices in the application sketched above, the following lemma and Corollary 4 are formulated more generally for arbitrary sets $T \subseteq S$.

▶ **Lemma 3.** *Let $\mathcal{G}$ be a connected temporal graph with lifetime ℓ and let $S \in \mathcal{G}/Aut(\mathcal{G})$ be any orbit. Let $T \subseteq S$ with $p = (|S| - |T|)/|S|$ and $W = (u_1, u_2, \ldots, u_x)$ a temporal walk starting at time step t and ending at t' with $t, t' \in [\ell]$ and $u_1 \in S$ such that W visits k vertices of S. Then there exists a temporal walk W' starting at a vertex $u' \in S$ in time step t and ending at time step t' that visits at least pk vertices of $S \setminus T$.*

When restricting the possible start vertices where the temporal walk W' of Lemma 3 is allowed to start, we obtain the following corollary. In detail, we are given a set $X \subset S$ such that the walk W' is only allowed to begin at a vertex $u' \in X$. In Lemma 3, W' was allowed to start at any vertex of S. Our use case for the following corollary is that X is a set of vertices that can be reached faster, making them better candidates for start vertices when extending walks in the way sketched above Lemma 3.

▶ **Corollary 4.** *Let $\mathcal{G}$ be a connected temporal graph with lifetime ℓ and let $S \in \mathcal{G}/Aut(\mathcal{G})$ be any orbit. Let $T \subsetneq S$ and W a temporal walk starting at time step t and ending at t' with $t, t' \in [\ell]$ such that the first vertex of W is in S and such that W visits k different vertices of S. For any $X \subseteq S$ with $|X| > |T|$ there exists a temporal walk W' starting at a vertex $u' \in X$ at time step t and ending at time step t' that visits at least $(c-1)/c \cdot k$ vertices of $S \setminus T$, with $c = |X|/|T|$.*

4 Upper Bounds for TEXP

A common approach to build a temporal walk for TEXP is to use Lemma 1, i.e., to construct a (large) set X of reachable vertices so that an unseen vertex v of the current walk is in the set X and the walk can then be extended by v. We are interested in exploring the vertices of one orbit quickly, as this will be useful for TRP in Section 5 where the agents try to meet in one orbit, and for TEXP because we can explore a temporal graph orbit by orbit. Therefore, we want to find walks visiting many vertices of one orbit. Our approach is similar to the common approach mentioned above, and so we want to construct a (large) set X of reachable vertices, but now with the property that X is a subset of the orbit under consideration. To construct X, we show in Lemma 6 a kind of "reachability between orbits."

To describe this in more detail, we need the concept of so-called lanes. Intuitively, lanes are defined for a set of vertices that are all contained in some single orbit, and give us knowledge about the vertices that are quickly reachable while only using orbit boundary edges in each time step. Using this concept of lanes we derive a first result for exploring a single large orbit with a temporal walk that spans $O((n^{5/3}+rn)\log n)$ time steps (Theorem 8). In the proof of that lemma we build the final temporal walk iteratively, by concatenating multiple smaller temporal walks. To make sure each new such small temporal walk visits a desired number of vertices not yet visited, we use Lemma 3, which – informally – lets us transform temporal walks that visit too many previously visited vertices into temporal walks that visit many previously unvisited vertices.

We follow this up with a more refined technique that considers the size of the orbit S one wants to explore as a parameter, but also uses the concept of lanes and walk transformations sketched above. It gives us an upper bound of $O(|S|^{1+\epsilon} + n \log |S|)$, for any constant $\epsilon > 0$. This result is formulated in Theorem 11. Finally, we use a repeated application of Theorem 11 to achieve an upper bound for TEXP of $O(rn^{1+\epsilon})$. We start with an auxiliary lemma that focuses on the orbit boundary edges between two orbits.

▶ **Lemma 5.** *Let G_t be the graph at time step t in a connected temporal graph $\mathcal{G}$ and $S, S' \in \mathcal{G}/Aut(\mathcal{G})$, and let G' be the subgraph of G_t that contains only orbit boundary edges. Then all vertices in S have the same degree in the bipartite graph $G'[S \cup S']$.*

To describe reachability between orbits, we have to introduce some extra notation. Let $\mathcal{G}$ be a temporal graph with lifetime ℓ and vertex set V and $S \in \mathcal{G}/\mathrm{Aut}(\mathcal{G})$ be an orbit. We call a *lane* $L_{t,t'}(X)$ with $X \subseteq S$ and $t, t' \in [\ell]$ the set of all vertices reachable from any $u \in X$ in G by any temporal walk W that only uses orbit boundary edges and starts in time step t and ends in time step at most t'. We write $L_{t,t'}(u)$ instead of $L_{t,t'}(\{u\})$. See Fig. 1 for some intuition.

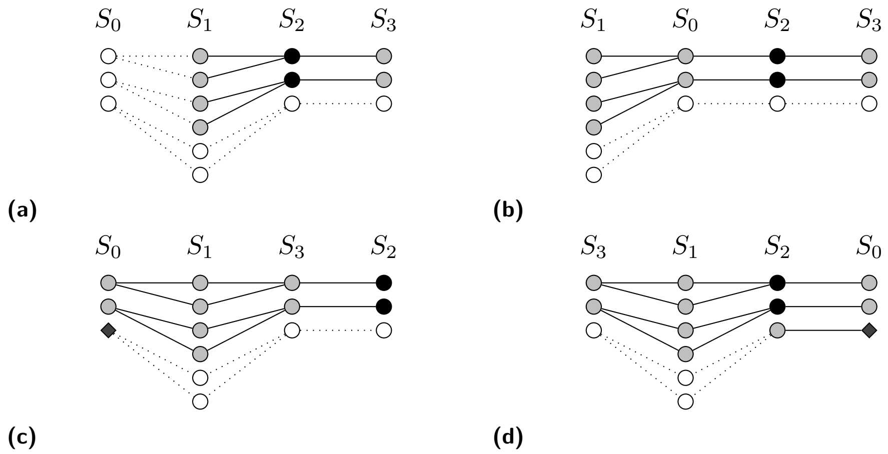

Figure 1 Visualization of the properties of a lane $L_{t,t+r}(X)$. Each vertex in the set X is colored black, and the set of reachable vertices in each time step is colored gray. Each figure represents one additional time step. In the second time step (Figure b) all vertices of the lane are reachable. In the third time step (Figure c) one vertex outside the lane must be reachable, for example the diamond shaped vertex. This is due to the fact that the temporal graph at hand is connected, and thus at least one additional vertex is reachable with every next time step. Here, one additional time step then suffices to reach a vertex of $S_2 \setminus X$ (Figure d).

The next lemma gives us a lower bound on the number of vertices of an orbit S' that can be reached from a subset X of the vertices of an orbit S within r time steps. Intuitively speaking, we show a lower bound on the number of vertices *reachable from orbit S in another orbit S'*. A simple consequence of the following lemma is that, from any start vertex in the temporal graph, at least one vertex in every orbit is reachable within r time steps.

▶ **Lemma 6** (Reachability between Orbits). *Let $\mathcal{G}$ be a connected temporal graph with lifetime ℓ and $S \in \mathcal{G}/Aut(\mathcal{G})$. For any $X \subseteq S$ and $S' \in \mathcal{G}/Aut(\mathcal{G})$ it holds that $|L_{t,t'}(X) \cap S'| \geq \lceil |X| \cdot |S'|/|S| \rceil$ for any $t \in [\ell]$ and $t' = t + r$, where $r = |\mathcal{G}/Aut(\mathcal{G})|$ is the orbit number.*

By using Lemma 1 and Lemma 6, we now bound the number of time steps needed to reach a set of h vertices within an orbit S.

▶ **Lemma 7.** *Let $\mathcal{G}$ be a connected temporal graph with lifetime ℓ and vertex set V. Let $S \in \mathcal{G}/Aut(\mathcal{G})$ and let $r = |\mathcal{G}/Aut(\mathcal{G})|$ be the orbit number. For any $h \leq |S|$, start vertex $u \in S$ and start time t, there exists a set $X \subset S$ with $|X| = h$ such that we can reach any vertex in X in at most $O(\min\{h \cdot n/|S|, hr\} + r)$ time steps. That is, for every vertex u' of X, we have a temporal walk starting at u at time step t and ending at u' at time step t' with $t' - t = O(\min\{h \cdot n/|S|, hr\} + r)$.*

Next we present Theorem 8, which states an upper bound for visiting all vertices of a given orbit S. The rough idea used in the proof is that we iteratively build the final temporal walk W by concatenating smaller temporal walks. In each step of the iteration, a small temporal walk W' is first constructed via Lemma 7 to visit a subset of the vertices of S, which are not necessarily unvisited, but such that the size of the subset is at least a certain threshold value. Using Lemma 3 we find an automorphism σ that we apply to W' to obtain a temporal walk W_σ that visits many unvisited vertices of S. We then extend W via this transformed walk W_σ (see Fig. 2 for a sketch of the proof idea). In this way we can explore all vertices of a large orbit S faster than by repeated application of Lemma 1. The key to obtain a good bound on the number of time steps required is to find a good value for the number of vertices of S visited by each small temporal walk.

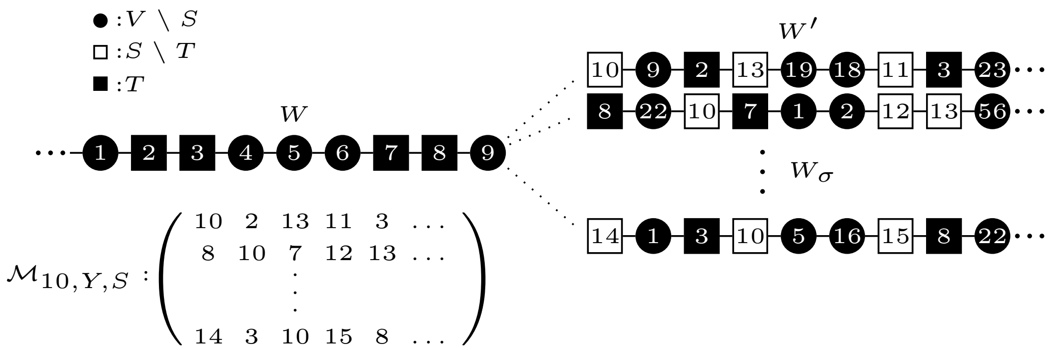

Figure 2 The construction scheme of Theorem 8. W is the temporal walk constructed so far. We aim to extend the walk with a walk W_σ that visits many vertices of $S \setminus T$, where T is the set of vertices of orbit S that we have already visited. To find W_σ, we construct W' and the automorphism matrix (bottom left) for the vertex with label 10 (the start vertex of W'), the set Y of vertices of S visited by W', and the entire orbit S as the set of possible start vertices of W_σ. One of the rows in the matrix then gives us an automorphism σ that, when applied to W', yields the desired walk W_σ.

▶ **Theorem 8.** *Let $\mathcal{G}$ be a temporal graph with lifetime ℓ and vertex set V. Take $S \in \mathcal{G}/Aut(\mathcal{G})$ and $r = |\mathcal{G}/Aut(\mathcal{G})|$ the orbit number. For any $t \in [\ell]$ there exists a temporal walk W starting at time step t that visits all vertices of S and ends at time step t' with $t' - t = O((n^{5/3} + rn)\log n)$.*

The following lemma is concerned with visiting a fraction $1/c$ of the vertices of a given orbit S with a temporal walk. One significant contribution to the number of time steps required by the temporal walk constructed in Theorem 8 is the use of Lemma 3. Roughly speaking, Lemma 3 provides a temporal walk that visits a large number of unvisited vertices, but with the caveat that every vertex of S can potentially be the start vertex of this transformed walk (instead of restricting the potential start vertices for the walk to a smaller subset, which might be reachable more quickly). The consequence of this is that for each such transformation we require, we must plan a "buffer" of n time steps to ensure that all vertices of S are reachable by the time step in which the transformed walk starts (Lemma 1). Corollary 4 provides a "trade-off" for this: a decrease in the set of possible start vertices of the transformed walk W_σ decreases the number of previously unvisited vertices W_σ visits, but also decreases the number of time steps required to reach the first vertex of W_σ. Using this property we construct a recursive algorithm that visits a fraction of the vertices of S quickly instead of applying the iterative construction of Theorem 8. In our recursive construction, the walks we concatenate shrink with each recursive call. If we were to use Lemma 3 during this, we would have an additional n time steps with each recursive call. Instead, Corollary 4 lets us reduce the number of possible start vertices dramatically. The time span required by this walk is then not dependent on n, but dependent on $|S|$ and r (the orbit number), and thus is especially useful for exploring smaller orbits. This can then be used iteratively to construct a temporal walk that visits all vertices of S, which we in turn use to visit all vertices V by visiting all orbits one after the other.

▶ **Lemma 9.** *Let $\mathcal{G}$ be a connected temporal graph with lifetime ℓ and vertex set V. Let $S \in \mathcal{G}/Aut(\mathcal{G})$ and let $r = |\mathcal{G}/Aut(\mathcal{G})|$ be the orbit number. For any $t \in [\ell]$ and any $u \in S$ there exists a temporal walk W that starts at vertex u in time step t and visits a fraction $1/c$ (for any $1 < c < |S|$) of the vertices of S such that W spans $O(rc(|S|/c)^{\phi(c)} \log |S|)$ time steps, with $\phi(c) = 1/(\log f(c))$ and $f(c) = (1 + (c-1)/c)$.*

▶ **Corollary 10.** *Let $\mathcal{G}$ be a temporal graph with lifetime ℓ and vertex set V. Let $S \in \mathcal{G}/Aut(\mathcal{G})$ and $r = |\mathcal{G}/Aut(\mathcal{G})|$ be the orbit number. For any $t \in [\ell]$, any $u \in S$, and any fixed $\epsilon > 0$, there exists a temporal walk W that starts at vertex u in time step t and visits some constant fraction $\alpha < 1$ of the vertices of S such that W spans $O(r|S|^{1+\epsilon})$ time steps.*

We now present an improved version of Theorem 8 for exploring a whole orbit.

▶ **Theorem 11.** *Let $\mathcal{G}$ be a temporal graph with lifetime ℓ and vertex set V. Let $S \in \mathcal{G}/Aut(\mathcal{G})$ and $r = |\mathcal{G}/Aut(\mathcal{G})|$ be the orbit number. For any $t \in [\ell]$, any $u \in V$, and any fixed $\epsilon > 0$, there exists a temporal walk W that starts at vertex u in time step t and visits all vertices of S such that W spans $O(r|S|^{1+\epsilon} + n \log |S|)$ time steps.*

By using the theorem above repeatedly for each orbit, we get a temporal walk for the whole temporal graph.

▶ **Corollary 12.** *Let $\mathcal{G}$ be a temporal graph with lifetime ℓ and vertex set V. For any fixed $\epsilon > 0$, there exists a temporal walk W that spans $O(rn^{1+\epsilon})$ time steps and visits all vertices of V, where $r = |\mathcal{G}/Aut(\mathcal{G})|$ is the orbit number.*

5 Upper Bound for TRP

Using Theorem 11 we show that TRP can be solved by constructing a walk that spans (asymptotically) the same number of steps as a walk for exploring an arbitrary single orbit. The idea is that the two agents identify an orbit in which they meet, and then the first agent moves to this orbit, and after n time steps the second agent starts exploring this orbit. For this the agents must be able to independently identify the same orbit, for which we introduce some additional notation. We extend the definition of isomorphism to temporal graphs as follows. Let $\mathcal{G}, \mathcal{H}$ be two temporal graphs with lifetime ℓ and vertex sets $V_{\mathcal{G}}$ and $V_{\mathcal{H}}$, respectively. We call a bijection $\theta : V_{\mathcal{G}} \to V_{\mathcal{H}}$ a *temporal isomorphism* if θ is an isomorphism from G_t to H_t for each $t \in [\ell]$ (and thus also an isomorphism from G to H, which denote the underlying graphs of $\mathcal{G}$ and $\mathcal{H}$, respectively). If clear from the context, we say *isomorphism* instead of *temporal isomorphism*.

We define an *integer coloring* as a coloring of the vertices in the vertex set V of a temporal graph $\mathcal{G}$ (with the colors being integer values). The assigned colors induce a partial order $\prec_{\mathcal{P}}$ on the vertex set V such that, for all vertices $u, v \in V$ that are assigned colors c_u and c_v, respectively, with $c_u \neq c_v$ it holds that $u \prec_{\mathcal{P}} v$ if $c_u < c_v$. The idea is now that the two agents compute the same integer coloring of the given temporal graph $\mathcal{G}$ with the property that two vertices $u, v \in V$ are assigned the same color c if and only if $u, v \in S$, with S some orbit of $\mathcal{G}/\mathrm{Aut}(\mathcal{G})$. The agents then meet at the smallest orbit, breaking ties via the coloring.

Note that, since the agents do not have access to consistent labels of the vertices in V, they are unable to distinguish between two vertices $u, v \in S$ with S being an orbit. Intuitively, the two agents a_1 and a_2 view $\mathcal{G}$ as different temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, such that $\mathcal{G}_1 \cong \mathcal{G} \cong \mathcal{G}_2$. A natural idea is for the agents to pick a smallest orbit for their meeting, but the challenge is how to ensure that the agents pick the same orbit if there are multiple equal-size orbits that all have the smallest size. Therefore, in the proof of the following lemma we let the agents iterate over all possible temporal graphs until they find a graph $\mathcal{H}$ with $\mathcal{G}_1 \cong \mathcal{H} \cong \mathcal{G}_2$. Then both agents compute an integer coloring for $\mathcal{H}$ as outlined in the previous paragraph. This coloring is translated to a coloring of $\mathcal{G}_1$ by agent a_1 and to a coloring of $\mathcal{G}_2$ by agent a_2 via isomorphism functions, which are independently computed by the agents.

▶ **Lemma 13.** *Let $\mathcal{G}$ be a temporal graph with vertex set V and lifetime ℓ and let a_1, a_2 be two label-oblivious agents. There exists a pair of programs (p_1, p_2) assigned to a_1 and a_2, respectively, such that each agent computes the same integer coloring of V and such that two vertices $u, v \in V$ have the same color exactly if u, v are in the same orbit of $\mathcal{G}/Aut(\mathcal{G})$.*

We can now easily construct an algorithm for TRP. The agents simply meet in a smallest orbit, breaking ties via the integer coloring (Lemma 13). The first agent moves to said orbit, then the second agent searches the orbit for the first agent. Note that the smallest orbit has size at most n/r, where r is the orbit number. Thus, the bound on the number of time steps provided by Theorem 11 becomes $O(r(n/r)^{1+\epsilon} + n\log(n/r)) = O(n^{1+\epsilon})$, and we obtain the following upper bound for TRP.

▶ **Theorem 14.** *Let $\mathcal{G}$ be a temporal graph with lifetime ℓ and a_1, a_2 two label-oblivious agents. For any fixed $\epsilon > 0$, there exists a pair of programs (p_1, p_2) assigned to a_1, a_2, respectively, such that the two agents are guaranteed to meet after $O(n^{1+\epsilon})$ time steps.*

6 Lower Bounds for TEXP and TRP

We start this section with a simple lower bound for TEXP, which is a fairly straightforward adaptation of the known lower bound of $\Omega(n^2)$ time steps [27]. Following that, we give a lower bound of $\Omega(n \log n)$ time steps for TRP. For this we describe the construction of a temporal graph that is connected and has only a single orbit. We then show how an adversary can choose the starting positions of the two agents that want to meet in order to delay their meeting. Intuitively, the graph we create is a cycle that changes repeatedly after some number of steps. By our construction, the adversary can make sure that after every change of the graph, the two agents are placed far away from each other. In the end, we also show that the resulting lower bound for TRP yields a corresponding lower bound for TEXP.

▶ **Lemma 15.** *For any $1 \leq r \leq n$, there exist n-vertex instances of TEXP with orbit number r that require $\Omega(rn)$ time steps to be explored.*

▶ **Theorem 16.** *For any two agents a_1 and a_2 with arbitrary deterministic programs, there exist instances of TRP where the agents require $\Omega(n \log n)$ time steps to meet.*

▶ **Corollary 17.** *There exist connected temporal graphs $\mathcal{G}$ with vertex set V, lifetime ℓ and a single orbit such that all temporal walks W require $\Omega(n \log n)$ time steps to visit all vertices of V.*

The lower bounds of Theorem 16 and Corollary 17 can be adapted to temporal graphs with r orbits for any constant r as follows: Use the construction from the proof of Theorem 16, but instead of letting the graph G_t in each time step be a single cycle C_i, let G_t contain $2r-1$ copies of C_i, and for each vertex u of C_i connect all copies of u by a path P_u (starting with the vertex u in the first copy of C_i and ending with the vertex u in the $(2r-1)$-th copy of C_i). The resulting temporal graph has r orbits: The vertices of the "middle" copy of C_i form one orbit, and the vertices in the two copies of C_i that have distance k from the middle copy, for $1 \leq k \leq r-1$, also form an orbit. Let $n' = n/r$ denote the number of vertices in one copy of C_i. By the arguments in the proof of Theorem 16, it takes $\Omega(n' \log n')$ time steps for the two agents to reach a location in the same path P_u, and thus TRP requires $\Omega(n' \log n')$ time steps. As $n' = n/r$ and r is a constant, this gives a lower bound of $\Omega(n \log n)$ for TRP. The lower bound of $\Omega(n \log n)$ time steps for exploration of temporal graphs with orbit number r for any constant r then follows as in the proof of Corollary 17.

7 Conclusions & Future Work

In this work, we looked at temporal graphs where agents know the complete information of the temporal graph ahead of time. In this clairvoyant setting, we studied the temporal exploration problem (TEXP) and showed how to bound the exploration time of a temporal graph using the structural graph property of the number of orbits of the automorphism group of the temporal graph. Additionally, we formalized the problem of asymmetric rendezvous in this setting as the temporal rendezvous problem (TRP) and showed how to adapt our ideas for TEXP to solve TRP quickly. For both TEXP and TRP we provided lower bounds such that the gap between upper and lower bounds is $O(n^\epsilon)$ for any fixed $\epsilon > 0$. There are several ways in which our work can be extended. One line of research for both problems is to reduce the gap between the lower and upper bounds by improving either of them. A second line of work is to study the symmetric variant of rendezvous in the given setting and see if something can be said about it. Another interesting situation to explore is when multiple agents are used to explore the temporal graph (and also if multiple agents need to perform temporal rendezvous) and how much faster solutions in these scenarios might be. Lastly, a possible avenue of research is to study the structural properties provided by automorphism groups and how they can be used to tackle other problems that concern temporal graphs.

References

1 Eric Aaron, Danny Krizanc, and Elliot Meyerson. Dmvp: foremost waypoint coverage of time-varying graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 29–41. Springer, 2014. doi:10.1007/978-3-319-12340-0_3.

2 Eric Aaron, Danny Krizanc, and Elliot Meyerson. Multi-robot foremost coverage of time-varying graphs. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 22–38. Springer, 2014. doi:10.1007/978-3-662-46018-4_2.

3 Duncan Adamson, Vladimir V. Gusev, Dmitriy Malyshev, and Viktor Zamaraev. Faster exploration of some temporal graphs. In *1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SAND.2022.5.

4 Eleni C. Akrida, Jurek Czyzowicz, Leszek Gąsieniec, Łukasz Kuszner, and Paul G. Spirakis. Temporal flows in temporal networks. *Journal of Computer and System Sciences*, 103:46–60, 2019. doi:10.1016/j.jcss.2019.02.003.

5 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. doi:10.1016/j.jcss.2021.04.001.

6 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107:108–123, 2020. doi:10.1016/j.jcss.2019.08.002.

7 Steve Alpern. The rendezvous search problem. *SIAM Journal on Control and Optimization*, 33(3):673–683, 1995. doi:10.1137/S0363012993249195.

8 Steve Alpern. Rendezvous search: A personal perspective. *Operations Research*, 50(5):772–795, 2002. doi:10.1287/opre.50.5.772.363.

9 Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous*, volume 55 of *International Series in Operations Research & Management Science*. Springer Science & Business Media, 2006. doi:10.1007/b100809.

10 Steven Alpern. Hide and seek games. In *Seminar, Institut für höhere Studien, Wien*, volume 26, 1976.

11 K. Balasubramanian. Symmetry groups of chemical graphs. *International Journal of Quantum Chemistry*, 21(2):411–418, 1982. doi:10.1002/qua.560210206.

12 Fabian Ball and Andreas Geyer-Schulz. How symmetric are real-world graphs? A large-scale study. *Symmetry*, 10(1), 2018. doi:10.3390/sym10010029.

13 Subhash Bhagat and Andrzej Pelc. Deterministic rendezvous in infinite trees. *CoRR*, abs/2203.05160, 2022. doi:10.48550/arXiv.2203.05160.

14 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. doi:10.1016/j.ipl.2018.10.016.

15 John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976.

16 Marjorie Bournat, Swan Dubois, and Franck Petit. Gracefully degrading gathering in dynamic rings. In *Stabilization, Safety, and Security of Distributed Systems: 20th International Symposium, SSS 2018, Tokyo, Japan, November 4–7, 2018, Proceedings 20*, pages 349–364. Springer, 2018. doi:10.1007/978-3-030-03232-6_23.

17 Benjamin Merlin Bumpus and Kitty Meeks. Edge exploration of temporal graphs. *Algorithmica*, 85(3):688–716, 2023. doi:10.1007/s00453-022-01018-7.

18 Arnaud Casteigts. Efficient generation of simple temporal graphs up to isomorphism. Github repository, 2020. URL: https://github.com/acasteigts/STGen.

19 Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *Int. J. Parallel Emergent Distributed Syst.*, 27(5):387–408, 2012. doi:10.1080/17445760.2012.668546.

20 Jurek Czyzowicz, Adrian Kosowski, and Andrzej Pelc. How to meet when you forget: Log-space rendezvous in arbitrary graphs. In *Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '10, pages 450–459, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1835698.1835801.

21 Shantanu Das, Giuseppe Di Luna, Linda Pagli, and Giuseppe Prencipe. Compacting and grouping mobile agents on dynamic rings. In *International Conference on Theory and Applications of Models of Computation*, pages 114–133. Springer, 2019. doi:10.1007/978-3-030-14812-6_8.

22 Giuseppe Antonio Di Luna. Mobile agents on dynamic graphs. *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, pages 549–584, 2019. doi:10.1007/978-3-030-11072-7_20.

23 Giuseppe Antonio Di Luna, Paola Flocchini, Linda Pagli, Giuseppe Prencipe, Nicola Santoro, and Giovanni Viglietta. Gathering in dynamic rings. *Theoretical Computer Science*, 811:79–98, 2020. doi:10.1016/j.tcs.2018.10.018.

24 Yoann Dieudonné, Andrzej Pelc, and Vincent Villain. How to meet asynchronously at polynomial cost. In *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, pages 92–99, 2013. doi:10.1137/130931990.

25 Konstantinos Dogeas, Thomas Erlebach, Frank Kammer, Johannes Meintrup, and William K. Moses Jr au2. Exploiting automorphisms of temporal graphs for fast exploration and rendezvous, 2023. arXiv:2312.07140.

26 Jessica Enright, Kitty Meeks, George B. Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. *Journal of Computer and System Sciences*, 119:60–77, 2021. doi:10.1016/j.jcss.2021.01.007.

27 Thomas Erlebach, Michael Hoffmann, and Frank Kammer. On temporal graph exploration. *J. Comput. Syst. Sci.*, 119:1–18, 2021. doi:10.1016/j.jcss.2021.01.005.

28 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 für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.141.

29 Thomas Erlebach and Jakob T. Spooner. A game of cops and robbers on graphs with periodic edge-connectivity. In *46th International Conference on Current Trends in Theory and Practice of Informatics (SOFSEM 2020)*, volume 12011 of *Lecture Notes in Computer Science*, pages 64–75. Springer, 2020. doi:10.1007/978-3-030-38919-2_6.

30 Thomas Erlebach and Jakob T. Spooner. Exploration of k-edge-deficient temporal graphs. *Acta Informatica*, 59(4):387–407, 2022. doi:10.1007/s00236-022-00421-5.

31 Thomas Erlebach and Jakob T. Spooner. Parameterized temporal exploration problems. In *1st Symposium on Algorithmic Foundations of Dynamic Networks, SAND 2022, March 28-30, 2022, Virtual Conference*, volume 221 of *LIPIcs*, pages 15:1–15:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SAND.2022.15.

32 Paola Flocchini. *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*. Springer, 2019. doi:10.1007/978-3-030-11072-7.

33 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. As time goes by: Reflections on treewidth for temporal graphs. In *Treewidth, Kernels, and Algorithms: Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, pages 49–77. Springer International Publishing, 2020. doi:10.1007/978-3-030-42071-0_6.

34 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020. doi:10.1016/j.tcs.2019.03.031.

35 Chris Godsil and Gordon F. Royle. *Algebraic Graph Theory*. Number Book 207 in Graduate Texts in Mathematics. Springer, 2001. doi:10.1007/978-1-4613-0163-9.

36 Thekla Hamm, Nina Klobas, George B. Mertzios, and Paul G. Spirakis. The complexity of temporal vertex cover in small-degree graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):10193–10201, June 2022. doi:10.1609/aaai.v36i9.21259.

37 David Ilcinkas, Ralf Klasing, and Ahmed Mouhamadou Wade. Exploration of constantly connected dynamic graphs based on cactuses. In *International Colloquium on Structural Information and Communication Complexity*, pages 250–262. Springer, 2014. doi:10.1007/978-3-319-09620-9_20.

38 David Ilcinkas and Ahmed Mouhamadou Wade. Exploration of the t-interval-connected dynamic graphs: the case of the ring. In *International Colloquium on Structural Information and Communication Complexity*, pages 13–23. Springer, 2013. doi:10.1007/978-3-319-03578-9_2.

39 Ulrich Knauer and Kolja Knauer. *Algebraic graph theory: morphisms, monoids and matrices*, volume 41. Walter de Gruyter GmbH & Co KG, 2019. doi:10.1515/9783110617368.

40 Evangelos Kranakis, Danny Krizanc, and Euripides Marcou. *The mobile agent rendezvous problem in the ring*. Springer Nature, 2022. doi:10.1007/978-3-031-01999-9.

41 Evangelos Kranakis, Danny Krizanc, and Sergio Rajsbaum. Mobile agent rendezvous: A survey. In *Structural Information and Communication Complexity*, pages 1–9. Springer Berlin Heidelberg, 2006. doi:10.1007/11780823_1.

42 Josef Lauri and Raffaele Scapellato. *Topics in Graph Automorphisms and Reconstruction*. Cambridge University Press, 2016. doi:10.1017/CBO9781316669846.

43 Gianluca De Marco, Luisa Gargano, Evangelos Kranakis, Danny Krizanc, Andrzej Pelc, and Ugo Vaccaro. Asynchronous deterministic rendezvous in graphs. *Theoretical Computer Science*, 355(3):315–326, 2006. doi:10.1016/j.tcs.2005.12.016.

44 Andrea Marino and Ana Silva. Königsberg sightseeing: Eulerian walks in temporal graphs. In *Combinatorial Algorithms*, pages 485–500. Springer International Publishing, 2021. doi:10.1007/978-3-030-79987-8_34.

45 Andrea Marino and Ana Silva. Coloring temporal graphs. *Journal of Computer and System Sciences*, 123:171–185, 2022. doi:10.1016/j.jcss.2021.08.004.

46 George B. Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. Computing Maximum Matchings in Temporal Graphs. In *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, volume 154 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2020.27.

47 George B. Mertzios, Hendrik Molter, and Viktor Zamaraev. Sliding window temporal graph coloring. *Journal of Computer and System Sciences*, 120:97–115, 2021. doi:10.1016/j.jcss.2021.03.005.

48 Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Math.*, 12(4):239–280, 2016. doi:10.1080/15427951.2016.1177801.

49 Othon Michail and Paul G. Spirakis. Traveling salesman problems in temporal graphs. *Computer Science (MFCS)*, 2014.

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

51 Othon Michail, Paul G. Spirakis, and Michail Theofilatos. Beyond rings: Gathering in 1-interval connected graphs. *Parallel Processing Letters*, 31(04):2150020, 2021. doi:10.1142/S0129626421500201.

52 Nils Morawietz, Carolin Rehs, and Mathias Weller. A Timecop's Work Is Harder Than You Think. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 71:1–71:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.71.

53 Fukuhito Ooshita and Ajoy K. Datta. Brief announcement: feasibility of weak gathering in connected-over-time dynamic rings. In *Stabilization, Safety, and Security of Distributed Systems: 20th International Symposium, SSS 2018, Tokyo, Japan, November 4–7, 2018, Proceedings 20*, pages 393–397. Springer, 2018. doi:10.1007/978-3-030-03232-6_27.

54 Andrzej Pelc. Deterministic rendezvous in networks: A comprehensive survey. *Networks*, 59(3):331–347, 2012. doi:10.1002/net.21453.

55 Andrzej Pelc. Deterministic rendezvous algorithms. In *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, pages 423–454. Springer, 2019. doi:10.1007/978-3-030-11072-7_17.

56 Andrzej Pelc. Deterministic rendezvous algorithms. *CoRR*, abs/2303.10391, 2023. doi:10.48550/arXiv.2303.10391.

57 Philipp Plamper, Oliver J. Lechtenfeld, Peter Herzsprung, and Anika Groß. A temporal graph model to predict chemical transformations in complex dissolved organic matter. *Environmental Science & Technology*, 2023. doi:10.1021/acs.est.3c00351.

58 Nicola Santoro. Time to change: On distributed computing in dynamic networks (keynote). In *19th International Conference on Principles of Distributed Systems, OPODIS 2015, December 14-17, 2015, Rennes, France*, volume 46 of *LIPIcs*, pages 3:1–3:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.OPODIS.2015.3.

59 C. Shannon. Presentation of a maze solving machine. In *Trans. 8th Conf. Cybernetics: Circular, Causal and Feedback Mechanisms in Biological and Social Systems (New York, 1951)*, pages 169–181, 1951.

60 Masahiro Shibata, Naoki Kitamura, Ryota Eguchi, Yuichi Sudo, Junya Nakamura, and Yonghwan Kim. Partial gathering of mobile agents in dynamic tori. In *2nd Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.SAND.2023.2.

61 Masahiro Shibata, Yuichi Sudo, Junya Nakamura, and Yonghwan Kim. Partial gathering of mobile agents in dynamic rings. In *Stabilization, Safety, and Security of Distributed Systems: 23rd International Symposium, SSS 2021, Virtual Event, November 17–20, 2021, Proceedings 23*, pages 440–455. Springer, 2021. doi:10.1007/978-3-030-91081-5_29.

62 Shadi Taghian Alamouti. *Exploring temporal cycles and grids.* PhD thesis, Concordia University, 2020.

63 Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. The complexity of finding small separators in temporal graphs. *Journal of Computer and System Sciences*, 107:72–92, 2020. doi:10.1016/j.jcss.2019.07.006.