Brief Announcement: Exploring Word-Representable Temporal Graphs

Duncan Adamson
School of Computer Science, University of St Andrews, UK

Abstract

Word-representable graphs are a subset of graphs that may be represented by a word w over an alphabet composed of the vertices in the graph. In such graphs, an edge exists if and only if the occurrences of the corresponding vertices alternate in the word w. We generalise this notion to temporal graphs, constructing timesteps by partitioning the word into factors (contiguous subwords) such that no factor contains more than one copy of any given symbol. With this definition, we study the problem of *exploration*, asking for the fastest schedule such that a given agent may explore all n vertices of the graph. We show that if the corresponding temporal graph is connected in every timestep, we may explore the graph in $2\delta n$ timesteps, where δ is the lowest degree of any vertex in the graph. In general, we show that, for any temporal graph represented by a word of length at least $n(2dn + d)$, with a connected underlying graph, the full graph can be explored in $2dn$ timesteps, where d is the diameter of the graph.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Temporal Graphs, Word-Representable Graphs

Digital Object Identifier 10.4230/LIPIcs.SAND.2025.22

Related Version *Full Version*: https://arxiv.org/abs/2502.07496 [1]

1 Introduction

Word-Representable graphs, first introduced by Kitaev and Pyatkin [12], are a subset of graphs that may be represented in a concise manner via a word (or string) of symbols. In these graph, the vertices are represented by the alphabet, while edges are defined by the structure of the word. See [10] and [11] for an overview of the main results on word-representable graphs.

Much work on word-representable graphs has focused on determining which such graphs are representable [12, 3, 9]. Such work has shown that a large number of graph classes are representable by word graphs, including three colourable graphs, circle graphs, and crown graphs. As such, the study of problems on these graphs are non-trivial.

In this work, we extend the notion of word-representable graphs to *temporal graphs*. Temporal graphs are a generalisation of static graphs, replacing the single edge set with an ordered sequence of sets. Each edge set is used to define a *timestep* (also known as a snapshot), during which the only active edges are those in the corresponding timestep. Temporal graphs have attracted a large amount of attention in recent years, with a particular focus on reachability [4, 6, 13] and exploration [2, 5, 7, 8, 14].

Our Contributions. We present a set of results on the problem of exploring word-representable temporal graphs. This combines research on word-representable graphs, temporal graphs, and graph exploration. We provide two primary results. First, we show that for any word-representable temporal graph where each timestep is a fully connected graph, we can construct an exploration schedule requiring at most $2\delta n$ timesteps, where δ is the minimum degree of any vertex in the graph and n the number of vertices. Second, we show that for any word-representable temporal graph where the underlying graph is connected and

4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2025).
Editors: Kitty Meeks and Christian Scheideler; Article No. 22; pp. 22:1–22:6
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the word representation requires at least $n(2dn+d)$ symbols, we can construct an exploration schedule requiring $2dn$ timesteps, where d is the diameter of the graph, and n the number of vertices.

Related Work. While there is a large body of work on both word-representable graphs and exploration of temporal graphs, we highlight a small number of papers in both fields of particular relevance. Regarding word-representable graphs, we point first to the work of Kitaev and Pyatkin [12] who introduced word-representable graphs as a tool for semigroup theory. From the prespective of structural results on word-representable graphs, in [3], Courcelle shows that circle graphs are word-representable. Halldorson et al. [9] strengthen this by showing that comparability graphs and all 3-colourable graphs a word-representable, as well as presenting communication from Limouzy that cover graphs are word-representable. On the negative side, Halldorson et al. show that it is NP-hard to determine if a given triangle free graph is word-representable.On the exploration of temporal graphs, we first mention the landmark paper by Erlebach et al. [7], which established many of the key ideas behind temporal graph exploration. In particular, they show a general $O(n^2)$ upper bound on the length of the fastest exploration schedule for a single agent on an always connected temporal graph. This is complemented by schedules of length $O(n^{1.5}k^{1.5}\log n)$ for graphs with treewidth k, $O(n^{1.8}\log n)$ for planar graphs, $O(n)$ for cycles with one chord, and $O(n\log^3 n)$ for $2\times n$ grids. Building on this, Erlebach and Spooner showed in [8] that an always connected temporal graph in which each timestep is at most k-edges inactive can be explored in $O(kn\log n)$ timesteps.

2 Preliminaries

Graphs and Temporal Graphs. We define a *graph* G by a tuple containing a set V of vertices, and set $E \subseteq V \times V$ of edges, each a pair of vertices. When writing an edge explicitly as (v_i, v_j) we call v_i the *start point* and v_j the *end point* of the edge. We call two vertices $v_i, v_j \in V$ *neighbours* if $(v_i, v_j) \in E$. We denote by $N(v) = \{v' \in V \mid (v, v') \in E\}$ the set of neighbours of v.. The *degree* of a vertex $v \in V$, denoted $\Delta(v) = |N(v)|$, is equal to the number of neighbours of v. A *walk* in a graph G is an ordered sequence of edges, $e_1, e_2, \dots, e_k$ such that the end point of the i^{th} edge is the start point of the $(i+1)^{th}$ edge. The start point of a walk is the start point of the first edge of the walk, and the end point of a walk is the end point of the last edge of the walk. Two vertices, v_i and v_j, are *connected* if there exists some walk starting at v_i and ending at v_j. We denote the set of all walks in a graph G by $\mathcal{W}(G)$. The *length* of a walk $P = e_1, e_2, \dots, e_k$, denoted $|P|$ is the number of edges in the walk. A walk *visits* a vertex v if there exists at least one edge e such that $v \in e$. The *distance* between two vertices $v_i, v_j \in V$, denoted $\text{dist}(v_i, v_j)$ is the walk of minimum length with v_i as the start point and v_j as the end point. We define the *diameter* of a graph as $\max_{v_i,v_j \in V} \text{dist}(v_i, v_j)$.

A *temporal graph* $\mathcal{G}$ is a generalisation of a graph, defined over one set of vertices V, and T sets of edges, $E_1, E_2, \dots, E_T$. We refer to the graph formed from the vertex set V and the t^{th} edge set, E_t, as the t^{th} *timestep*, denoted $G_t = (V, E_t)$. The *lifetime* of a temporal graph $\mathcal{G} = (V, E_1, E_2, \dots, E_T)$ is equal to the number T of edge sets in $\mathcal{G}$. The *underlying graph* of a temporal graph $\mathcal{G} = (V, E_1, E_2, \dots, E_T)$, denoted $U(\mathcal{G})$, is the graph formed over the vertex set V and the union of all edge sets, giving $U(\mathcal{G}) = (V, \bigcup_{t\in[1,T]} E_t)$. A *temporal walk* is an extension of a walk, with each step being an edge-time pair, $(e_1, t_1)(e_2, t_2)\dots(e_k, t_k)$ such that, for every $i \in [1, k-1]$:

- the edge e_i is active in the timestep E_{t_i},
- the end point of e_i is the start point of e_{i+1}, and,
- $t_{i+1} > t_i$.

Additionally, e_k must be active in timestep t_k. Note that t_{i+1} may be greater than $t_i + 1$. In this case, assuming $e_i = (v_j, v_i)$, we say that an agent traversing this walk *waits* at v_i for $t_{i+1} - t_i - 1$ timesteps. A temporal walk $P = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$ visits a vertex v iff there is some edge e_i in the walk such that $v \in e_i$. The *length* of a temporal walk containing k edges is equal to t_k.

Words. An *alphabet* $\Sigma = \{1, 2, \dots, \sigma\}$ is a finite set of symbols. A *word* (also known as a *string*) w is a finite sequence of symbols from a given alphabet. We assume, for the remainder of this paper, that our alphabet Σ is defined over some set of integers $1, 2, \dots, \sigma$. The length of a word w, denoted $|w|$ is the number of letters in the word. For $i \in [1, |w|]$ let $w[i]$ denote the i^{th} letter of w. The set of all finite words over the alphabet Σ is denoted by Σ^*. Given $n \in \mathcal{N}_0$, let Σ^n denote all words in Σ^* exactly of length n. Given two words w, v, we denote by wv the word formed by the concatenation of w and v, i.e. the word such that $wv[i] = w[i]$, if $i \in [1, |w|]$ or $v[i - |w|]$ if $i \in [|w| + 1, |w| + |v|]$. Given a word $w \in \Sigma^*$ and natural $k \in \mathbb{N}_0$, let w^k denote the word formed by k copies of w, satisfying $|w^k| = k|w|$ and $w^k[i] = w[i \bmod |w|]$, $\forall i \in [1, k|w|]$. Let $\text{alph}(w) = \{\mathtt{a} \in \Sigma \mid \exists i \in [|w|] \text{ s.t. } w[i] = \mathtt{a}\}$ be the alphabet of w.

Given a pair of words $u, w \in \Sigma^*$ where $|u| \leq |w|$, u is a *subsequence* of w if there exists some set of indices $i_1, i_2, \dots, i_{|u|}$ such that $1 \leq i_1 < i_2 \dots < i_{|u|} \leq |w|$ such that $u = w[i_1]w[i_2] \dots w[i_{|u|}]$. Given a set of symbols $\mathcal{S} \subseteq \Sigma$ and word $w \in \Sigma^*$, the subsequence of w $\pi_{\mathcal{S}}(w)$ satisfies $\pi_{\mathcal{S}}(w) \in \mathcal{S}^*$, and, $\forall u \in \mathcal{S}^*$ either $|u| \leq |\pi_{\mathcal{S}}(w)|$ or u is not a subsequence of w.

Word-Representable (Temporal) Graphs. Given a word $w \in \Sigma^*$, the graph represented by w, $G(w)$, is constructed as follows. Let $V = (v_1, v_2, \dots, v_n)$ be a set of n vertices, each labelled uniquely by some symbol from $\Sigma = 1, 2, \dots, n$. Informally, $G(w)$ contains the edge (v_x, v_y) iff the symbols x and y *alternate* in w. A pair of symbols, $x, y \in \Sigma$ alternate in w if $\pi_{\{x,y\}}(w) \in \{(xy)^k, (xy)^k x, (yx)^k, (yx)^k y \mid k \in \mathbb{N}_0\}$. Thus, we define the edge set E as the set $\{(v_x, v_y) \mid \pi_{\{x,y\}}(w) \in \{(xy)^k, (xy)^k x, (yx)^k, (yx)^k y \mid k \in \mathbb{N}_0\}\}$. A graph G is *word-representable* if there exists some word w such that $G(w) = G$.

Given a word $w \in \Sigma^*$, the temporal graph represented by w, $TG(w)$, is constructed using $G(w)$ as a basis. To determine the timesteps, we introduce the set of indices $S_1, S_2, \dots, S_T \in [1, |w|]$ as the set of *start points* for each timestep. The value of each is computed recursively, with $S_1 = 1$, and S_i the index such that $w[S_i] \in \text{alph}(w[S_{i-1}, S_i - 1])$ and, $\forall S_{i'} \in [S_{i-1}, S_i - 1], w[S_i'] \notin \text{alph}(w[S_{i-1} + 1, S_{i'-1}])$.

With this set of indices, the timestep t is defined with regard to the symbols appearing in the factor $w[S_t, S_{t+1} - 1]$ (or, $w[S_T, |w|]$ for the final timestep). Formally, given some edge (v_x, v_y) in the edge set E of $G(w)$, the edge $(v_x, v_y) \in E_t$ iff either $x \in \text{alph}(w[S_t, S_{t+1} - 1])$ (resp., $x \in w[S_T, |w|]$) or $y \in \text{alph}(w[S_t, S_{t+1} - 1])$ (resp., $y \in w[S_T, |w|]$). We call any factor of w of the form $w[S_i, S_{i+1} - 1]$ a *timestep factor.*

3 Always Connected Graphs

In this section, we provide results on exploring *always connected graphs.* A temporal graph $\mathcal{G}$ is always connected if there exists, at every timestep, exactly one connected component containing all vertices in $\mathcal{G}$.

▶ **Lemma 1.** *Let $\mathcal{G} = (V, E_1, E_2, \ldots, E_T)$ be a word-representable temporal graph represented by the word w, such that $\mathcal{G}$ is connected in every timestep. Then, given any vertex $v_x \in V$, $x \in \text{alph}(w[S_t, S_{t+\Delta(v_x)+1} - 1])$, $\forall t \in [1, T - \Delta(x) - 2]$, and $x \in \text{alph}(w[S_{T-\Delta(v_x)-1}, |w|]$ where $S_1, S_2, \ldots, S_T$ are the set of start points of w.*

▶ **Corollary 2.** *Let $\mathcal{G} = (V, E_1, E_2, \ldots, E_T)$ be a word-representable temporal graph represented by the word w, such that $\mathcal{G}$ is connected in every timestep. Then the edge $(v_x, v_y) \in E_t \cup E_{t+1} \cup \cdots \cup E_{t+\min(\Delta(v_x),\Delta(v_y))}$, $\forall t \in [1, T - \min(\Delta(v_x), \Delta(v_y))]$.*

We now strengthen Lemma 1 and Corollary 2 to prove that each edge e in any word-representable always-connected temporal graph appears at least once in every set of $\delta + 1$ contiguous timesteps. The main idea is to propagate the constraints given implicitly in Corollary 2 across the graph. At a high level, we have that not only must some symbol x satisfy $x \in \text{alph}(w[S_t, S_{t+\Delta(v_x)+1} - 1]$, as given in Lemma 1, but also $x \in \text{alph}(w[S_t, S_{t+\min(\Delta(v_x),\Delta(v_y))+1} - 1]$ as indicated by Corollary 2 and formally stated in Lemma 3. More generally, however, x must appear at least once between each occurrence of y, for any y such that $v_y \in N(v_x)$. Thus, any bound on the number of occurrences of y can be propagated through the graph to all other symbols. By extension, we get a bound on the number of timesteps in which an edge may be absent. Lemma 3 formalises this idea.

▶ **Lemma 3.** *Let $\mathcal{G} = (V, E_1, E_2, \ldots, E_T)$ be a word-representable temporal graph represented by the word w, such that $\mathcal{G}$ is connected in every timestep. Then the edge $(v_x, v_y) \in E_t \cup E_{t+1} \cup \cdots \cup E_{t+\delta}$, $\forall t \in [1, T - \delta]$, where $\delta = \min_{v \in V} \Delta(v)$.*

We now present our main theorem for this section, showing that any word-representable always-connected temporal graph may be explored in $2\delta n$ timesteps. The high-level idea behind this proof is to first construct a spanning tree on the underlying graph, then use a walk P exploring this tree to derive a temporal walk exploring the temporal graph. We do so by following the same set of edges as P, waiting at each vertex at most δ timesteps for the next edge to become available. As P can contain at most $2n$ edges, our temporal walk requires at most $2\delta n$ timesteps, giving the algorithmic result.

▶ **Theorem 4.** *Let $\mathcal{G} = (V, E_1, E_2, \ldots, E_T)$ be a word-representable temporal graph represented by the word w, such that $\mathcal{G}$ is connected in every timestep. Then, $\mathcal{G}$ can be explored in $2\delta|V|$ timesteps starting at any vertex v_x, where $\delta = \min_{v \in V} \Delta(v)$.*

4 General Graphs

We now consider general word-representable temporal graphs. We assume, without loss of generality, that all graphs in the section contain a single component in the underlying graph, noting that any graph that does not satisfy this can not be explored. As in Section 3, we start with some structural results on the word representations of these graphs.

▶ **Lemma 5.** *Given a word-representable temporal graph $\mathcal{G}$ represented by the word w containing the edge (v_x, v_y). Then, $|\pi_{\{x\}}(w)| - 1 \leq |\pi_{\{y\}}| \leq |\pi_{\{x\}}(w)| + 1$.*

▶ **Lemma 6.** *Let $v_x, v_y \in V$ be a pair of vertices such that $\text{dist}(v_x, v_y) = d$. Then, $|\pi_{\{x\}}(w)| - d \leq |\pi_{\{y\}}(w)| \leq |\pi_{\{x\}}(w)| + d$.*

▶ **Lemma 7.** *Let $v_x, v_y \in V$ be a pair of vertices such that $\text{dist}(v_x, v_y) = d$. Further, let χ_i be the index of the i^{th} occurrence of x in w, and let γ_j be the index of the j^{th} occurrence of y in w. Then $\chi_{i-d} \leq \gamma_i \leq \chi_{i+d}$.*

▶ **Corollary 8.** *Given a word-representable temporal graph $\mathcal{G}$, represented by the word w, $U(\mathcal{G}) = (V, \bigcup_{t\in[1,d]} E_t)$, where d is the diameter of $U(\mathcal{G})$.*

From Lemmas 5, 6, 7 and Corollary 8, we have the main tools used to build our algorithmic result. At a high level, we use a similar technique to Theorem 4, building a temporal walk exploring the graph from a walk exploring the spanning tree of the underlying graph. Lemma 9 and Corollary 10 provide the last needed results for the approach, showing that each edge in the graph is inactive for at most d consecutive timesteps.

▶ **Lemma 9.** *Given a word-representable temporal graph $\mathcal{G}$, represented by the word w such that $U(G)$ has a diameter of d. Then, given any edge (v_x, v_y) active in timestep $t \in [1, T-d-1]$, $\exists t' \in [t+1, t+d+1]$ such that $(v_x, v_y) \in E_{t'}$.*

▶ **Corollary 10.** *Given a word-representable temporal graph $\mathcal{G}$, represented by the word w such that the underlying graph $U(\mathcal{G})$ has a diameter of d. Then, for any $t \in [1, T-d]$, $U(\mathcal{G}) = (V, \bigcup_{d'\in[0,d]} E_{t+d'})$.*

From Lemma 9 and Corollary 10, we now apply a similar technique to Theorem 4 to derive an algorithm for exploring word-representable temporal graphs of length at least $n(2dn + d)$ in $2dn$ timesteps. Informally, we construct a exploration of $U(\mathcal{G})$ by way of an Eulerian walk on a spanning tree of $U(\mathcal{G})$. As each edge is active at least once every d timesteps per Corollary 10, this may be converted into a temporal walk exploring $\mathcal{G}$ requiring at most $2dn$ timesteps.

▶ **Theorem 11.** *Let $\mathcal{G} = (V, E_1, E_2, \dots, E_T)$ be a word-representable temporal graph, represented by the word w such that $|w| \geq n(2dn + d)$, where d is the diameter of the underlying graph $U(\mathcal{G}) = (V, \bigcup_{t\in[T]} E_t)$. Then, $\mathcal{G}$ can be explored in at most $2dn$ timesteps.*

References

1. Duncan Adamson. Exploring word-representable temporal graphs, 2025. doi:10.48550/arXiv.2502.07496.
2. 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.
3. Bruno Courcelle. Circle graphs and monadic second-order logic. *Journal of Applied Logic*, 6(3):416–442, 2008. doi:10.1016/J.JAL.2007.05.001.
4. Argyrios Deligkas and Igor Potapov. Optimizing reachability sets in temporal graphs by delaying. *Inf. Comput.*, 285(Part):104890, 2022. doi:10.1016/J.IC.2022.104890.
5. Konstantinos Dogeas, Thomas Erlebach, Frank Kammer, Johannes Meintrup, and William K Moses Jr. Exploiting automorphisms of temporal graphs for fast exploration and rendezvous. *arXiv preprint arXiv:2312.07140*, 2023. doi:10.48550/arXiv.2312.07140.
6. 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.
7. Thomas Erlebach, Michael Hoffmann, and Michael Kammer. On temporal graph exploration. *Journal of Computer and System Sciences*, 119:1–18, 2021. doi:10.1016/J.JCSS.2021.01.005.
8. Thomas Erlebach and Jakob T. Spooner. Exploration of k-edge-deficient temporal graphs. In Anna Lubiw and Mohammad Salavatipour, editors, *Algorithms and Data Structures*, pages 371–384, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-83508-8_27.
9. Magnús M. Halldórsson, Sergey Kitaev, and Artem Pyatkin. Semi-transitive orientations and word-representable graphs. *Discrete Applied Mathematics*, 201:164–171, 2016. doi:10.1016/J.DAM.2015.07.033.

10 Sergey Kitaev. A comprehensive introduction to the theory of word-representable graphs. In *International Conference on Developments in Language Theory*, pages 36–67. Springer, 2017. doi:10.1007/978-3-319-62809-7_2.

11 Sergey Kitaev and Vadim Lozin. *Words and graphs*. Springer, 2015. doi:10.1007/978-3-319-25859-1.

12 Sergey Kitaev and Artem Pyatkin. On representable graphs. *Journal of Automata, Languages and Combinatorics*, 13(1):45–54, 2008. doi:10.25596/JALC-2008-045.

13 Kitty Meeks. Reducing reachability in temporal graphs: Towards a more realistic model of real-world spreading processes. In *Conference on Computability in Europe*, pages 186–195. Springer, 2022. doi:10.1007/978-3-031-08740-0_16.

14 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.