Information Processing Letters 142 (2019) 68–71

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On exploring always-connected temporal graphs of small pathwidth ☆

Hans L. Bodlaender [a,b,*,1], Tom C. van der Zanden [a]

[a] *Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands*
[b] *Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, the Netherlands*

ARTICLE INFO

Article history:
Received 1 August 2018
Received in revised form 24 October 2018
Accepted 24 October 2018
Available online 25 October 2018
Communicated by Marcin Pilipczuk

Keywords:
Graph algorithms
Computational complexity
Temporal graphs
Graph exploration
Pathwidth

ABSTRACT

We show that the Temporal Graph Exploration Problem is NP-complete, even when the underlying graph has pathwidth 2 and at each time step, the current graph is connected.

1. Introduction

Networks can change during time: roads can be blocked or built, friendships can wither or new friendships are formed, connections in a computer network can go down or be made available, etc. Temporal graphs can serve as a model for such changing networks.

A *temporal graph* is a finite sequence of graphs with the same vertex set, i.e., a temporal graph $\mathcal{G}$ is given by a series of graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, ..., $G_L = (V, E_L)$. Each i, $1 \le i \le L$ is called a *time step*. The graph G_i is called the *current graph at time step i*. Note that while the vertex set is the same at each time step, the set of edges can (and in general will) be different at the different time steps. The main intuition is that at time step i, only the edges in G_i exist and can be used.

The *underlying* graph is formed by taking the union of the graphs at the different time steps, i.e., for a temporal graph $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, ..., $G_L = (V, E_L)$, the *underlying graph* is the graph $(V, E_1 \cup E_2 \cup \cdots \cup E_L)$, so an edge exists in the underlying graph if and only if it exists in at least one time step.

In temporal graphs, we can define a *temporal walk*: we have an explorer who at time step 0 is at some s; at each time step i she can move over an edge in G_i or remain at her current location. More precisely, a temporal walk in temporal graph $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, ..., $G_L = (V, E_L)$ is a sequence of vertices $v_0, v_1, \ldots, v_K$ with $K \le L$, such that for each i, $1 \le i \le K$, $v_{i-1} = v_i$ or $\{v_{i-1}, v_i\} \in E_i$. We say that the walk is at v_{i-1} at the beginning of time step i, and at v_i at the end of time step i.

In this note, we study the complexity of a problem on temporal graphs: the Temporal Graph Exploration problem, as introduced by Michail and Spirakis [9]. In the Temporal Graph Exploration problem, we are given a tempo-

☆ Declarations of competing interest: none.

* Corresponding author at: Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands.

E-mail addresses: H.L.Bodlaender@uu.nl (H.L. Bodlaender), T.C.vanderZanden@uu.nl (T.C. van der Zanden).

1 The research of this author was partially supported by the *Networks* project, supported by the Netherlands Organization for Scientific Research N.W.O. under project no. 024.002.003.

https://doi.org/10.1016/j.ipl.2018.10.016

ral graph, a specified start vertex s, and an integer L, and ask if there is a temporal walk with L time steps that visits all vertices in the temporal graph at least once. A variant is when we require that the walk ends at the starting vertex s; we denote this by RTB TEMPORAL GRAPH EXPLORATION, with RTB the acronym of *return to base*. (See [1].)

Background It is easy to see that even if the graphs do not change over time, the TEMPORAL GRAPH EXPLORATION problem is NP-complete, as it contains HAMILTONIAN PATH as a special case (set $L = n - 1$.) Michail and Spirakis [9] introduced the problem, showed that it does not have a c-approximation, unless P = NP, and obtained approximation algorithms for several special cases.

Recently, Akrida et al. [1] studied the TEMPORAL GRAPH EXPLORATION problem when the underlying graph is a star $K_{1,r}$. Even when each edge exists in at most six time steps, the problem is NP-complete. We use the following of their results as starting point.

Theorem 1 *(Akrida et al. [1]).* RTB TEMPORAL GRAPH EXPLORATION *is NP-complete, when the underlying graph is a star, and each edge exists in at most six graphs G_i, and the start and end vertex is the center of the star.*

An important special case is when we require that at each time step, the current graph G_i is connected. We use the term *always-connected* to denote this case, i.e., a temporal graph $G_1, \ldots, G_L$ is always-connected, if and only if each G_i is a connected graph. Now, if the number of time steps L is sufficiently large compared to the number of vertices n, it is always possible to explore an always-connected temporal graph. Specifically, Erlebach et al. [5] showed that an always-connected temporal graph can be explored in $O(t^2 n\sqrt{n}\log(n))$ time steps, where t is the treewidth of the underlying graph. Similarly, if the underlying graph is a 2 by n grid then $O(n\log^3 n)$ time steps always suffice.

For more results, including special cases, approximation algorithms and inapproximability results, see [1,5,9], and see [8] for a survey.

Graphs of small treewidth and pathwidth and our contribution It is well known that problems that are intractable (e.g., NP-hard) on general graphs become easier (e.g., linear time solvable) when restricted to graphs of bounded treewidth (see e.g., [3, Chapter 7].) An example is HAMILTONIAN PATH, which can be solved in $O(2^{O(t)}n)$ time on graphs of treewidth t [2,4]. Unfortunately, these positive results appear not to carry over to temporal graphs: we show that the TEMPORAL GRAPH EXPLORATION problem is NP-hard for always-connected temporal graphs, even when the underlying graph has pathwidth 2 (and thus also treewidth 2); moreover, at each time step, the current graph is a tree. Our result builds upon the hardness result by Akrida 1, adding a construction that ensures the always-connectedness; a star has pathwidth one; applying our construction adds one to the pathwidth. One easily observes that the TEMPORAL GRAPH EXPLORATION problem for always-connected graphs with treewidth 1 is trivial: a connected graph with treewidth 1 is a tree, and thus, all graphs G_i are equal to each other, and one can easily compute in linear time the number of steps to explore this fixed tree.

Interestingly, there are other problems on temporal graphs that do become tractable when the treewidth is bounded. Specifically, Fluschnik et al. [6] showed that finding a small *temporal separator* becomes tractable when the underlying graph has bounded treewidth; the problem is NP-hard in general [7].

Some graph theoretic definitions The pathwidth of graphs was defined by Robertson and Seymour [11]. A *path decomposition* of a graph $G = (V, E)$ is a sequence of subsets (called *bags*) of V $(X_1, \ldots, X_r)$, such that $\bigcup_{1\le i\le r} X_r = V$, for all $\{v, w\} \in E$, there is an i with $v, w \in X_i$, and if $1 \le i_1 < i_2 < i_3 \le r$, then $X_{i_1} \cap X_{i_3} \subseteq X_{i_2}$. The *width* of a path decomposition $(X_1, \ldots, X_r)$ equals $\max_{1\le i\le r} |X_i| - 1$; the *pathwidth* of a graph G is the minimum width of a path decomposition of G. The pathwidth of a graph is an upper bound for its treewidth. (See e.g. [3, Chapter 7].)

$K_{1,r}$ is a star graph with $r + 1$ vertices, i.e., we have one vertex of degree r which is adjacent to the remaining r vertices, which have degree 1.

2. Hardness result

We now give our main result.

Theorem 2. *The* TEMPORAL GRAPH EXPLORATION PROBLEM *is NP-complete, even if each graph $G_i = (V, E_i)$ is a tree, and the underlying graph has pathwidth 2.*

Proof. We use a polynomial-time reduction from RTB TEMPORAL GRAPH EXPLORATION for graphs whose underlying graph is a star (see Theorem 1) to our problem. Suppose we have a temporal graph $\mathcal{K}_{1,n-1}$, whose underlying graph is a star, given by a series of subgraphs of $K_{1,n-1}$, $G_1 = (V, E_1)$, ..., $G_L = (V, E_L)$, and a start vertex s, which is the center of the star. We denote the vertices of $K_{1,n-1}$ by $v_0, \ldots, v_{n-1}$, with $s = v_0$.

We now build a new temporal graph, as follows. Set $Q = L \cdot (n + 3)$.

The vertex set of the new graph consists of V and $Q + 1$ new vertices. The new vertices will form a path. The new vertices are denoted $p_0, \ldots, p_Q$ and called *path vertices*; the vertices in V are called *star vertices*.

We now define a temporal graph $\mathcal{G}'$, given by a series of graphs G'_i, $1 \le i \le L + Q + 1$. G'_i has the following edges:

- For each i, the vertices $p_0, \ldots, p_Q$ form a path: we have edges $\{p_j, p_{j+1}\}$ for $1 \le j < Q$.
- If $i \le L$, all edges in G_i are also edges in G'_i.
- If $i \le L$, for each star vertex $v_j \in V$: if there is a connected component of G_i with vertex set W, such that $j = \min_{v_{j'} \in W} j'$, then we have an edge $\{v_j, p_{L\cdot(j+2)}\}$. I.e., v_j has the smallest index j over all vertices in the same connected component of G_i.
- If $i > L$, we have an edge from each star vertex $v_i \ne s$ to s, and an edge from s to p_0.

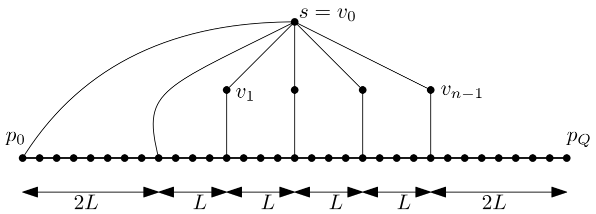

Fig. 1. Illustration to the proof of Theorem 2. Note that edges between path vertices are present at each time step; other edges are present in a subset of the time steps.

We look for a temporal walk that starts at s, visits all vertices, and takes $L + Q + 1$ time steps. See Fig. 1 for an illustration of the construction.

It is not hard to see that each G'_i is a tree. If $i \le L$, then G'_i is obtained by adding the path to G_i and one edge from the path to each connected component of G_i. If $i \ge L$, then G'_i is obtained taking a path and $K_{1,n}$ and adding an edge between a path and star vertex.

The idea behind the proof is that during the first L time steps, we explore the star vertices as normal, while the path serves to keep the graph connected but can not be explored. To explore the path vertices, we must make one single pass from p_0 to p_Q, as we do not have sufficient time steps to traverse either the section from p_0 to p_{2L-1} or that from p_{Q-2L+1} twice: traversing the edges between star vertices and path vertices (other than edge $\{s, p_0\}$) cannot contribute to a solution.

Lemma 3. *There is a temporal walk in $\mathcal{G}'$ that starts at s and visits all vertices in G' in $L + Q + 1$ time steps, if and only if there is a temporal walk in the temporal star $\mathcal{K}_{1,n-1}$ that starts at s, ends in s and visits all vertices in $\mathcal{K}_{1,n-1}$ in L time steps.*

Proof. First, suppose that there is a temporal walk in $\mathcal{K}_{1,n-1}$ that starts at s, ends at s and visits all vertices in at most L time steps. Then, we visit all vertices in $\mathcal{G}'$, by first making the temporal walk in the star, if necessary wait in s until the end of time step L and at time step $L+1$ move from s to p_0, and then visit all path vertices by traversing the path in the remaining Q time steps.

Suppose we have a temporal walk that starts at s and visits all vertices in $\mathcal{G}'$ in at most $L + Q + 1$ time steps.

Claim 4. *If we are at a path vertex p_i at the end of time step $\alpha \le L$, then $L < i < Q - L$.*

Proof. If we are at a path vertex p_i at the end of time step $\alpha \le L$, then we moved one or more times from a star vertex to a path vertex during the first α time steps. Consider the last of these moves, say that we moved at time step $\beta \le \alpha$ from a star vertex v_j to a path vertex $p_{j'}$; between time step $\beta + 1$ and α we stay at path vertices. As $\beta \le L$, the only path vertex adjacent to v_j in G_β is $p_{L\cdot(j+2)}$, so $j' = L \cdot (j + 2)$.

We can make less than L steps after reaching $p_{j'}$ until time step $\alpha \le L$, hence $j' - L < i < j' + L$. Now, $L = L \cdot (0 + 2) - L \le L \cdot (j+2) - L = j' - L < i < j' + L = L \cdot (j+2) + L \le L \cdot (n-1+2) + L = (n+2) \cdot L = Q - L$. This ends the proof of Claim 4. □

Claim 5. *At the end of time step L, we are in vertex s.*

Proof. Suppose not. Note that both p_0 and p_Q are not yet visited, by Claim 4. If we are at a star vertex $v_i \ne s$ at the end of time step L, then we can only visit the path vertices by first moving to s, then to p_0, and then visiting the path vertices in order; this costs one time step too many.

Suppose we are at a path vertex p_i at the end of time step L. First, suppose we visit p_0 before p_Q. Then, we must make at least i steps from p_i to p_0, and then Q steps from p_0 to p_Q. By Claim 4, $i > L$. So we need to make at least $i + Q > L + Q$ steps after time step L, which is a contradiction with the assumption that the walk takes $L + Q + 1$ time steps. Now, suppose we visit p_Q before p_0. Then we must make at least $Q - i$ steps from p_i to p_Q, and then Q steps from p_Q to p_0. By Claim 4, $Q - i > L$; we need to make at least $2Q - i > L + Q$ steps after time step L, again contradicting the assumption that the walk takes $L + Q + 1$ time steps. This ends the proof of Claim 5. □

Claim 6. *If at time step $i \le L$ we move from a star vertex v_i to a path vertex p_j, then the first star vertex visited after time step i is again v_i, and this move to v_i will be made before the end of time step L.*

Proof. By Claim 5, we must move to a star vertex before the end of time step L. If $p_{j'}$ is a neighbor of a star vertex and $j \ne j'$, then $p_{j'}$ is at least L steps on the path away from p_j, so we cannot reach $p_{j'}$ before time step L, hence we must move back to the star from p_j, and thus move to v_i. This ends the proof of Claim 6. □

Now, we can finish the proof of Lemma 3. Take from the walk in $\mathcal{G}'$ the first L time steps. Change this by replacing each move to a path vertex by a step where the explorer does not move. I.e., when the walk in $\mathcal{G}'$ moves from star vertex v_i to a path vertex, then we stay in v_i until the time step where the walk in G' moves back to the star — by Claim 6, this is a move to v_i. In this way, we obtain a walk in $\mathcal{K}_{1,n-1}$ that visits all vertices in L time steps. This ends the proof of Lemma 3. □

It remains to show that the underlying graph has pathwidth 2. If we remove s from the underlying graph, then we obtain a caterpillar: a graph that can be obtained by taking a path, and adding vertices of degree one, adjacent to a path vertex. These have pathwidth 1 [10]; now add s to all bags and we obtain a path decomposition of the underlying graph of $\mathcal{G}'$ of width 2. □

A minor variation of the proof gives also the following result.

Theorem 7. *The* RTB TEMPORAL GRAPH EXPLORATION PROBLEM *is NP-complete, even if each graph $G_i = (V, E_i)$ is a tree, and the underlying graph has pathwidth 2.*

Proof. Modify the proof of Theorem 2 as follows: add one time step; the current graph in the last time step has an edge, from p_Q to s, with additional edges added to make the graph into a tree (e.g., we could take the graph from the previous time step, add the edge (p_Q, s) and break the thus created cycle by removing one (other) edge). □

3. Conclusions

In this note, we showed that the Temporal Graph Exploration Problem is NP-complete, even when we require that at each time step, the graph is connected (i.e., we have an 'always-connected temporal graph'), or more specifically a tree, and the underlying graph (i.e., the graph where an edge exists whenever it exists for at least one time step) has pathwidth 2, and hence treewidth 2. This contrasts many other results for graphs of bounded treewidth, including a polynomial time algorithm for finding small temporal separators for graphs of small treewidth [6].

For always-connected temporal graphs, the case that the treewidth is 1 becomes trivial (as this deletes all temporal effects). For temporal graphs without the condition of always-connectedness, the Temporal Graph Exploration Problem is NP-complete for graphs of treewidth 1 by the results of Akrida et al. [1]. Interesting open cases are when the underlying graph is outerplanar, or an almost tree, i.e., can be obtained by adding one edge to a tree, and the temporal graph is always-connected.

References

[1] E.C. Akrida, G.B. Mertzios, P.G. Spirakis, The temporal explorer who returns to the base, arXiv:1805.04713 [abs], 2018.

[2] H.L. Bodlaender, M. Cygan, S. Kratsch, J. Nederlof, Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth, Inf. Comput. 243 (2015) 86–111.

[3] M. Cygan, F.V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.

[4] M. Cygan, S. Kratsch, J. Nederlof, Fast Hamiltonicity checking via bases of perfect matchings, J. ACM 65 (3) (2018) 1–46.

[5] T. Erlebach, M. Hoffmann, F. Kammer, On temporal graph exploration, in: Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, ICALP 2015, Part I, in: Lecture Notes in Computer Science, vol. 9134, Springer, 2015, pp. 444–455.

[6] T. Fluschnik, H. Molter, R. Niedermeier, P. Zschoche, Temporal graph classes: a view through temporal separators, arXiv:1803.00882 [abs], 2018, Extended abstract to appear in Proceedings 44th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2018.

[7] D. Kempe, J. Kleinberg, A. Kumar, Connectivity and inference problems for temporal networks, J. Comput. Syst. Sci. 64 (4) (2002) 820–842.

[8] O. Michail, An introduction to temporal graphs: an algorithmic perspective, Internet Math. 12 (4) (2016) 239–280.

[9] O. Michail, P.G. Spirakis, Traveling salesman problems in temporal graphs, Theor. Comput. Sci. 634 (2016) 1–23.

[10] A. Proskurowski, J.A. Telle, Classes of graphs with restricted interval models, Discrete Math. Theor. Comput. Sci. 3 (4) (1999) 167–176.

[11] N. Robertson, P.D. Seymour, Graph minors. I. Excluding a forest, J. Comb. Theory, Ser. B 35 (1983) 39–61.