

On Exploring Temporal Graphs of Small Pathwidth

Hans L. Bodlaender* Tom C. van der Zanden†

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.

In this note, we study the complexity of a problem on temporal networks: the TEMPORAL GRAPH EXPLORATION problem. Recently, Akrida et al. [1] showed that this problem is NP-complete, even when the underlying graph is a star. An important special case, studied by Erlebach et al. [5], is when at each point in time, the current graph is connected. This case is trivial when the underlying graph is a tree; we show that it is already NP-complete when the underlying graph has pathwidth 2.

A *temporal graph* \mathcal{G} is given by a series of graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, \dots , $G_L = (V, E_L)$, each with the same vertex set, but the set of edges can be different at different time steps. At time step i , only the edges in E_i exist and can be used. Each i , $1 \leq i \leq L$ is called a *time step*, G_i is

*Department of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands. Email: H.L.Bodlaender@uu.nl, and Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, the Netherlands. The research of this author was partially supported by the *Networks* project, supported by the Netherlands Organization for Scientific Research N.W.O.

†Department of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands. Email: T.C.vanderZanden@uu.nl

the *current graph* at time i . The *underlying graph* is formed by taking the union of the graphs at the different time steps. I.e., if we have L time steps, and graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, \dots , $G_L = (V, E_L)$, the underlying graph is $(V, E_1 \cup E_2 \cup \dots \cup E_L)$, so an edge exists in the underlying graph if it exists in at least one time step. Many graph properties can be studied in the setting of temporal graphs; this note looks at the problem of *exploring* the graph.

In temporal graphs, we can define a *temporal walk*: we have an explorer who at time step 1 is at a specified vertex s ; at each time step i she can move over an edge in G_i or remain at her current location. In the TEMPORAL GRAPH EXPLORATION problem, we are given a temporal graph and a starting vertex s , and are asked if there exists a temporal walk starting at s that visits all vertices within a given time L . 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].)

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

An important special case is when we require that at each time step, the current graph G_i is connected. Now, if the time L is sufficiently large compared to the number of vertices n , it is always possible to explore the graph. Specifically, Erlebach et al. [5] showed that in this case, the 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.

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.*

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

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, even when the underlying graph has pathwidth 2 (and thus also treewidth 2), and at each point in time, the graph is a tree, and thus connected.

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].

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, \dots, X_r), such that $\bigcup_{1 \leq i \leq r} X_i = V$, for all $\{v, w\} \in E$, there is an i with $v, w \in X_i$, and if $1 \leq i_1 < i_2 < i_3 \leq r$, then $X_{i_2} \subseteq X_{i_1} \cap X_{i_3}$. The *width* of a path decomposition (X_1, \dots, X_r) equals $\max_{1 \leq i \leq r} |X_i| - 1$; 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 reduction from RTB TEMPORAL GRAPH EXPLORATION for star graphs. Suppose we have a temporal star $\mathcal{K}_{1,n-1}$, given by a series of subgraphs of $K_{1,n-1}$, $G_1 = (V, E_1), \dots, G_L = (V, E_L)$, and a start vertex s , which is the center of the star. We denote the vertices of $K_{1,n}$ by v_0, \dots, 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. These will form a path. The new vertices are denoted p_0, \dots, 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 , $1 \leq i \leq L'$. G'_i has the following edges:

- For each i , the vertices p_0, \dots, p_Q form a path: we have edges $\{p_j, p_{j+1}\}$ for $1 \leq j < Q$.
- If $i \leq L$, all edges in G_i are also edges in G'_i .
- If $i \leq L$, for each star vertex $v_j \in V$: if v_j is the lowest numbered vertex in a connected component of G_i , we have an edge $\{v_j, p_{L \cdot (j+2)}\}$.
- If $i > L$, we have an edge from each star vertex $v_i \neq s$ to s , and an edge from s to p_0 .

It is not hard to see that each G'_i is a tree. If $i \leq L$, 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 $i \geq L$, then G'_i is obtained taking a path and $K_{1,n}$ and adding an edge between a path and star vertex.

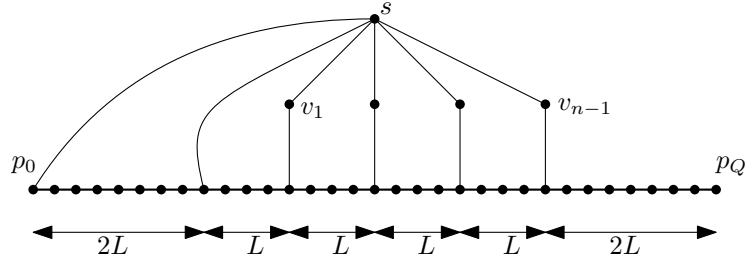


Figure 1: Illustration to the proof of Theorem 2. Note that edges between path vertices are present at each time step; other edges are present in a subset of the time steps.

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 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 at most $L + Q + 1$ 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 at most 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 $L + 1$ move from s to p_1 , 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 $\alpha \leq L$, then $L < i < Q - L$.*

Proof. If we are at a path vertex p_i at the end of time step $\alpha \leq L$, 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 $\beta \leq \alpha$ from a star vertex v_j to a path vertex $p_{j'}$; between time step $\beta + 1$ and α we stay at path vertices. We have that $j' = L \cdot (j + 2)$, by construction of the temporal graph. We can make less than L steps after reaching $p_{j'}$ until time step $\alpha \leq L$, hence $j' - L < i < j' + L$. Now, $L = L \cdot (0 + 2) - L \leq L \cdot (j + 2) - L = j' - L < i < j' + L = L \cdot (j + 2) + L \leq L \cdot (n - 1 + 2) + L = (n + 2) \cdot L = Q - L$. \square

Claim 5. *At the end of time step L , we must be 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 $v_i \neq s$ 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 . 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 : in total $i + Q > L + Q$ steps; contradiction. 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 : in total $2Q - i > L + Q$ steps; this is again a contradiction. \square

Claim 6. *If we move at time step $i \leq L$ 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 $j \neq j'$, then $p_{j'}$ is at least L steps on the path away from p_j , so we cannot reach $p_{j'}$ before time L , hence we must move back to the star from p_j , and thus move to v_i . \square

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 \mathcal{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. \square

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. \square

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 one edge, from p_Q to s . \square

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, 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].

If we require that the graph is connected at each time step, the case that the treewidth is 1 becomes trivial (as this deletes all temporal effects). 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.

References

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

- [2] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86 – 111, 2015.
- [3] M. Cygan, F. V. Fomin, L. Kowalik, D. Loksthanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [4] M. Cygan, S. Kratsch, and J. Nederlof. Fast Hamiltonicity checking via bases of perfect matchings. *Journal of the ACM*, 65(3):12:1–12:46, 2018.
- [5] T. Erlebach, M. Hoffmann, and F. Kammer. On temporal graph exploration. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, ICALP 2015, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 444–455. Springer, 2015.
- [6] T. Fluschnik, H. Molter, R. Niedermeier, and P. Zschoche. Temporal graph classes: A view through temporal separators. *arXiv*, abs/1803.00882, 2018. Extended abstract to appear in *Proceedings 44th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2018*.
- [7] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.
- [8] O. Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4):239–280, 2016.
- [9] O. Michail and P. G. Spirakis. Traveling salesman problems in temporal graphs. *Theoretical Computer Science*, 634:1–23, 2016.
- [10] A. Proskurowski and J. A. Telle. Classes of graphs with restricted interval models. *Discrete Mathematics & Theoretical Computer Science*, 3(4):167–176, 1999.
- [11] N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B*, 35:39–61, 1983.