Algorithmica (2023) 85:688–716
https://doi.org/10.1007/s00453-022-01018-7

Edge Exploration of Temporal Graphs

Benjamin Merlin Bumpus[1] · **Kitty Meeks**[2]

Received: 15 October 2021 / Accepted: 26 July 2022 / Published online: 22 August 2022

Abstract

We introduce a natural temporal analogue of Eulerian circuits and prove that, in contrast to the static case, it is NP-hard to determine whether a given temporal graph is temporally Eulerian even if strong restrictions are placed on the structure of the underlying graph and each edge is active at only three times. However, we do obtain an FPT-algorithm with respect to a new parameter called *interval-membership-width* which restricts the times assigned to different edges; we believe that this parameter will be of independent interest for other temporal graph problems. Our techniques also allow us to resolve two open questions of Akrida, Mertzios and Spirakis [CIAC 2019] concerning a related problem of exploring temporal stars.

Keywords Temporal graphs · Dynamic networks · Parameterized complexity · Width measures

1 Introduction

Many real-world problems can be formulated and modeled in the language of graph theory. However, real-world networks are often not *static*. They change over time and their edges may appear or disappear (for instance friendships may change over time in a social network). Such networks are called *dynamic* or *evolving* or *temporal* and their

An extended abstract of the present paper appeared in the proceedings of IWOCA '21 [1], containing statements of most of the results in Sections 3-5. This paper provides full proofs of these results, as well as some additional material in Section 6.

✉ Benjamin Merlin Bumpus
benjamin.merlin.bumpus@gmail.com

Kitty Meeks
kitty.meeks@glasgow.ac.uk

[1] Mathematics and Computer Science, TU Eindhoven, Groene Loper MetaForum Building, PO Box 513, Eindhoven 5600 MB, North Brabant, Netherlands

[2] School of Computing Science, University of Glasgow, Sir Alwyn Williams Building, Lilybank Gardens, Glasgow G12 8RZ, Scotland, UK

structural and algorithmic properties have been the subject of active study in recent years [2–6]. Some of the most natural and most studied topics in the theory of temporal graphs are temporal walks and paths (in which consecutive edges appear at increasing times), and the corresponding notions of temporal reachability [7–14]. Related to these notions is the study of explorability of a temporal graph which asks whether it is possible to visit all vertices or edges of a temporal graph via some temporal walk.

Temporal vertex-exploration problems (such as temporal variants of the Travelling Salesman problem) have already been thoroughly studied [15–17]. In contrast, here we focus on temporal *edge*-exploration and specifically we study *temporally Eulerian graphs*. Informally, these are temporal graphs admitting a temporal circuit (again in which edges appear at increasing times on a traversal of the circuit) that visits every edge at exactly one time (i.e. a temporal circuit that yields an Euler circuit in the underlying static graph). We also consider a closely-related problem in which every edge must be visited exactly twice.

Deciding whether a static graph is Eulerian is a prototypical example of a polynomial time solvable problem. In fact this follows from Euler's characterization of Eulerian graphs dating back to the 18th century [18]. In contrast, here we show that, unless P $=$ NP, a characterization of this kind cannot exist for temporal graphs. In particular, we show that deciding whether a temporal graph is *temporally Eulerian* is NP-complete even if strong restrictions are placed on the structure of the underlying graph and each edge is active at only three times. This complements a result obtained simutaneously and independently by Marino and Silva [19], which shows that this problem is NP-complete even when each edges is active at only two times, but without the same restrictions on the structure of the underlying graph.[1]

The existence of problems that are tractable on static graphs, but NP-complete on temporal graphs is well-known [3, 6, 15, 20]. In fact, there are examples of problems whose temporal analogues remain hard even on trees [15, 20]. Since many of the most studied graph parameters (such as treewidth and feedback vertex/edge number) are bounded on trees, it is clear that requiring boundedness of these parameters on the underlying input graphs for such problems will not be enough to give tractability. This motivates the development of parameters that take into account the temporal structure of the input, as well as the structure of the underlying graph. Some measures of this kind (such as temporal variants of feedback vertex number [10] and treewidth [21]) have already been studied. Unfortunately we find that these parameters are of no use to us since the problems we consider here remain NP-complete even when these measures are bounded by constants on the underlying static graph. To overcome these difficulties, we introduce a new purely-temporal parameter called *interval-membership-width*. Parameterizing by this measure we find that the problem of determining whether a temporal graph is temporally Eulerian is in FPT.

To understand the intuition behind this new parameter, consider the problem of determining whether a temporal graph in which every edge appears at exactly one time is temporally Eulerian. If two edges are active at the same time, the answer is clearly no; otherwise there is only one possible order in which we could traverse the

[1] Marino and Silva also studied other analogues of Euler circuits in temporal graphs, with different restrictions on the re-use of edges. Our notion of temporally Eulerian graphs corresponds, in their terminology, to graphs admitting an *Eulerian trail*.

edges in increasing order of times, so it suffices to check whether this gives rise to an Eulerian circuit in the underlying graph. More generally, the same argument shows that it is easy to decide whether a temporal graph is temporally Eulerian whenever each edge e is assigned a "private" interval of times, that is to say that no other edge is active at any time between the first and last appearance of e: again, this gives a single candidate order in which the edges can be traversed. The parameter interval-membership-width generalises this notion further, by quantifying the extent to which the set of intervals defined by the first and last appearance of each edge can overlap. The value of the parameter therefore does *not* depend on the structure of the underlying static graph (other than its number of edges), but is instead influenced only by the temporal structure. We believe that interval-membership-width will be a parameter of independent interest for other temporal graph problems: a dynamic programming approach similar to the one we use in Sect. 4 will lead to fixed-parameter-tractability whenever the crucial information we need to store about a partial solution at time t relates only to those edges that are active either at, or both before and after time t.

It turns out that our study of temporally Eulerian graphs is closely related to a temporal variant of the Travelling Salesman Problem concerning the exploration of temporal stars via a temporal circuit which starts at the center of the star and which visits all leaves (equivalently, each edge must be traversed exactly twice in the circuit). Akrida, Mertzios and Spirakis introduced this problem and proved it to be NP-complete even when the input is restricted to temporal stars in which every edge has at most k appearances for all $k \geq 6$ [15]. Although they also showed that the problem is polynomial-time solvable whenever each edge of the input temporal star has at most 3 appearances, they left open the question of determining the hardness of the problem when each edge has at most 4 or 5 appearances. We resolve this open problem in the course of proving our results about temporally Eulerian graphs. Combined with Akrida, Mertzios and Spirakis' results, this gives a complete dichotomy: their temporal star-exploration problem is in P if each edge has at most 3 appearances and is NP-complete otherwise.

As a potential "island of tractability", Akrida, Mertzios and Spirakis proposed to restrict the input to their temporal star-exploration problem by requiring that there exists some k such that, for each edge in the input graph, consecutive pairs in the set of times at which it is active differ by exactly k. Using our new notion of interval-membership-width we are able to show that this restriction does indeed yield tractability parameterized by the maximum number of temporal appearances of any edge (thus partially resolving their open problem). Furthermore, applying this same line of thought (i.e. studying inputs with evenly spaced edge-times) to the problem of determining whether a temporal graph is temporally Eulerian allows us to achieve a similar result.

Given the success of parameterizations by interval-membership-width when it comes to temporal edge-exploration problems, we also explore the applicability of this parameter to the related questions of vertex exploration and reachablity. It turns out that the parameter interval-membership-width is not so well-suited to problems in which we need to know which *vertices* (rather than which edges) have been used or reached at a particular time in the evolution of the temporal graph. We illustrate this intuition with a concrete example of a vertex-reachability problem (MinReachDelete) that remains

NP-hard even on classes of bounded interval-membership-width. This motivates us to introduce a "vertex-variant" of our measure called *vertex-interval-membership-width*. This new measure – which is lower-bounded by interval-membership-width – is more algorithmically powerful: for a vertex-reachability problem that remains hard even on temporal graphs with unit interval-membership-width, parameterizing by the vertex-variant of our measure allows us to achieve membership in FPT. Indeed, we show that MinReachDelete is in FPT when parameterized by vertex-interval-membership-width. Moreover, since the dissemination of a preliminary version of this paper, vertex-interval-membership-width has also been used to give FPT-algorithms for both a temporal version of the Firefighter problem [22] and the problem of counting paths in temporal graphs [23].

Outline. We fix notation and provide background definitions in Sect. 2. We prove our hardness results in Sect. 3. Sect. 4 contains the definition of interval-membership-width as well as our FPT algorithms parameterized by this measure. In Sect. 5 we show that Akrida, Mertzios and Spirakis' temporal star-exploration problem is in FPT parameterized by the maximum number of appearances of any edge in the input whenever the input temporal star has evenly spaced times on all edges. We also show a similar result for our temporally Eulerian problem. In Sect. 6 we introduce a vertex-variant of interval-membership-width and we discuss its algorithmic applications to vertex-exploration and reachability problems. Finally we provide concluding remarks and open problems in Sect. 7.

2 Background and Notation

For any graph-theoretic notation not defined here, we refer the reader to Diestel's textbook [24]; similarly, for any terminology in parameterized complexity, we refer the reader to the textbook by Cygan et al. [25].

The formalism for the notion of dynamic or time-evolving graphs to which we adhere originated from the work of Kempe, Kleinberg, and Kumar [12] (building on an earlier work of Berman [26]). Formally, if τ is a function $\tau : E(G) \to 2^{\mathbb{N}}$ mapping edges of a graph $G = (V(G), E(G))$ to sets of integers, then we call the pair $\mathcal{G} := (G, \tau)$ a *temporal graph* (this is a widely used generalization of the formulation in [12] which only allows one time to be assigned to each edge). We shall assume all temporal graphs to be finite and simple in this paper.

For any edge e in G, we call the set $\tau(e)$ the *time-set* of e. For any time $t \in \tau(e)$ we say that e is *active* at time t and we call the pair (e, t) a *time-edge*. The set of all edges active at any given time t is denoted $E_t(G, \tau) := \{e \in E(G) : t \in \tau(e)\}$. The greatest time Λ appearing in the time-set of any edge is said to be the *lifetime* of the graph, that is, $\Lambda := \max_{e\in E(G)} \max \tau(e))$. While it is possible to define temporal graphs with infinite lifetime, here we only consider temporal graphs whose lifetime is finite.

In a temporal graph there are two natural notions of walk: one is the familiar notion of a walk in static graphs and the other is a truly temporal notion where we require consecutive edges in walks to appear at non-decreasing times. Formally, given vertices x and y in a temporal graph $\mathcal{G}$, a *strict temporal* (x, y)*-walk* is a sequence $W = (e_1 =$

$xx', t_1), \ldots, (e_n = y'y, t_n)$ of time-edges such that $e_1, \ldots, e_n$ is a walk in G starting at x and ending at y and such that $t_1 < t_2 < \ldots < t_n$ (hereafter, since we will assume all temporal walks to be strict, we will simply refer to them as "temporal walks"). If $n > 1$, we denote by $W - (e_n, t_n)$ the temporal walk $(e_1, t_1), \ldots, (e_{n-1}, t_{n-1})$. We call a temporal (x, y)-walk *closed* if $x = y$.

Recall that an Euler circuit in a static graph G is a circuit $e_1 \ldots, e_m$ which traverses every edge of G exactly once. In this paper we are interested in a natural temporal analogue of this notion (called an *Eulerian Trail* by Marino and Silva [19]).

Definition 1 A *temporal Eulerian circuit* in a temporal graph (G, τ) is a closed temporal walk $(e_1, t_1), \ldots, (e_m, t_m)$ such that $e_1 \ldots, e_m$ is an Euler circuit in the underlying static graph G. If there exists a temporal Eulerian circuit in (G, τ), then we call (G, τ) *temporally Eulerian*.

Recall from the discussion in the introduction that, if (G, τ) is a temporal graph in which every edge appears at exactly one time (as considered in the original model of [12]), then we can determine whether (G, τ) is temporally Eulerian in time linear in $|E(G)|$. This observation suggests that the number of times per edge is relevant to the complexity of the associated decision problem, which we state as follows.

TEMPEULER(k)
Input: A temporal graph (G, τ) where $|\tau(e)| \leq k$ for every edge e in the graph G.
Question: Is (G, τ) *temporally Eulerian*?

As we mentioned in Sect. 1, here we show that TEMPEULER(k) is related to an analogue of the Travelling Salesman problem on temporal stars [15]. This problem (denoted as STAREXP(k)) was introduced by Akrida, Mertzios and Spirakis [15]. It asks whether a given temporal star (S_n, τ) (where S_n denotes the n-leaf star) with at most k times on each edge admits a closed temporal walk starting at the center of the star and which visits every leaf of S_n. We call such a walk an *exploration* of (S_n, τ). A temporal star that admits an exploration is called *explorable*. Formally we have the following decision problem.

STAREXP(k)
Input: A temporal star (S_n, τ) where $|\tau(e)| \leq k$ for every edge e in the star S_n.
Question: Is (S_n, τ) explorable?

No temporal star (be it explorable or not) can be temporally Eulerian (since the underlying static graph isn't). However, since any temporal star is explorable if and only if it admits a temporal circuit which visits every edge of the star exactly *twice*, the two notions are obviously rather closely related.

3 Hardness of Temporal Edge Exploration

This section is devoted to showing that TEMPEULER(k) is NP-complete for all k at least 3 (Corollary 4) and that STAREXP(k) is NP-complete for all k at least 4 (Corollary 2). This last result resolves an open problem of Akrida, Mertzios and Spirakis which asked to determine the complexity of STAREXP(4) and STAREXP(5) [15].

To show that STAREXP(4) is NP-hard, we provide a reduction from the 3-COLORING problem (see, for instance, Garey and Johnson [27] for a proof of NP-completeness of 3-COLORING) which asks whether an input graph G is 3-colorable.

> **3-COLORING**
> **Input:** A finite simple graph G.
> **Question:** Does G admit a proper 3-coloring (i.e. a coloring $c : V(G) \to \{1, 2, 3\}$ such that $c(u) \neq c(v)$ for all $uv \in E(G)$)?

Throughout, for an edge e of a temporal star (S_n, τ), we call any pair of times $(t_1, t_2) \in \tau(e)^2$ with $t_1 < t_2$ a *potential visit of e*. We say that e is *visited at* (t_1, t_2) in a temporal walk if the walk proceeds from the center of the star along e at time t_1 and then back to the center at time t_2. We say that two potential visits (t_1, t_2) and (t'_1, t'_2) of two edges e and e' are *in conflict* with one another (or that "there is a conflict between them") if there exists some time t such that

$$t_1 \le t \le t_2 \text{ and } t'_1 \le t \le t'_2.$$

Note that a complete set of potential visits (one visit for each edge of the star) which has no pairwise conflicts is in fact an exploration.

Theorem 1 STAREXP(4) *is* NP-*hard.*

Proof Take any 3-COLORING instance G with vertices $\{x_1, \ldots, x_n\}$. We will construct a STAREXP(4) instance (S_p, τ) (where $p = n + 3m$) from G.

Defining $\mathbf{S_p}$. The star S_p is defined as follows: for each vertex x_i in G, we make one leaf node in A_p which is connected to the center by an edge e_i while, for each edge $x_i x_j$ with $i < j$ in G, we make three leaf nodes in S_p which are connected to the center by edges e^0_{ij}, e^1_{ij} and e^2_{ij} respectively.

Defining τ. For $i \in [n]$ and any integer $\xi \ge 0$, let t^ξ_i be the integer

$$t^\xi_i := 2in^2 + 2\xi(n+1) \tag{1}$$

and take any edge $x_j x_k$ in G with $j < k$. Using the times defined in Eq. (1), we then define $\tau(e_i)$ and $\tau(e^\xi_{jk})$ as

$$\tau(e_i) := \left\{t^0_i, t^1_i, t^2_i, t^3_i\right\} \text{ and} \tag{2}$$

$$\tau(e^\xi_{jk}) := \left\{t^\xi_j + 2k - 1, \quad t^\xi_j + 2k, \quad t^\xi_k + 2j - 1, \quad t^\xi_k + 2j\right\}. \tag{3}$$

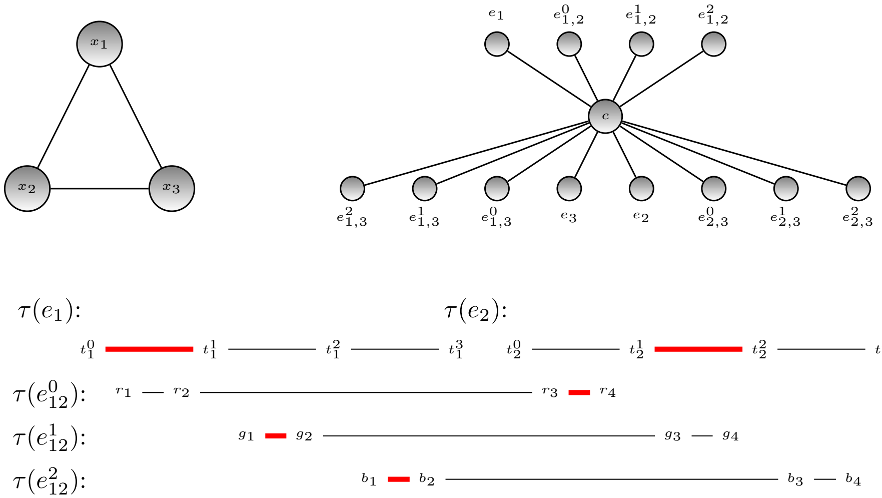

Fig. 1 Top left: K^3; we assume the coloring $x_i \mapsto i-1$. Top right: star constructed from K^3. Bottom: times (and corresponding intervals) associated with the edges e_1, e_2 and $e^0_{1,2}$, $e^1_{1,2}$, $e^2_{1,2}$ (time progresses left-to-right and intervals are not drawn to scale). We write r_1, r_2, r_3, r_4 as shorthand for the entries of $\tau(e^0_{1,2})$ (similarly, for $i \in [4]$, we write g_i and b_i with respect to $\tau(e^1_{1,2})$ and $\tau(e^2_{1,2})$). The red and thick intervals correspond to visits defined by the coloring of the K^3 (Color figure online)

Note that the elements of these sets are written in increasing order (see Fig. 1).

Intuitively, the times associated to each edge $e_i \in E(S_p)$ corresponding to a vertex $x_i \in V(G)$ (Eq. (2)) encode the possible colorings of x_i via the three possible starting times of a potential visit of e_i. The three edges e^0_{ij}, e^1_{ij} and e^2_{ij} corresponding to some $x_i x_j \in E(G)$ are instead used to force the colorings to be proper in G. That is to say that, for a color $\xi \in \{0, 1, 2\}$, the times associated with the edge e^{ξ}_{ij} (Eq. (3)) will prohibit us from entering e_i at its ξ-th appearance and also entering e_j at its ξ-th appearance (i.e. coloring x_i and x_j the same color).

Observe that the first two times in $\tau(e^{\xi}_{jk})$ lie within an interval given by consecutive times in $\tau(e_j)$ and that the same holds for the last two times in $\tau(e^{\xi}_{jk})$ with respect to $\tau(e_k)$ (see Fig. 1). As a consequence it is immediate that for $1 \le j < k$ and $\xi \ge 0$, we have:

$$t^{\xi}_j \;<\; t^{\xi}_j + 2k - 1 < \; t^{\xi}_j + 2k < \; t^{\xi+1}_j \tag{4}$$

Given this set-up, we now show that G is a yes instance if and only if (S_p, τ) is.

Suppose (S_p, τ) **is explorable.** Note first that, if e_i is visited at (t, t'), then we must have $t \in \{t^0_i, t^1_i, t^2_i\}$ (as there is no possible visit entering e_i at time t^3_i). We can therefore define a coloring (to be shown proper)

$$c : V(G) \to \{0, 1, 2\}$$

taking each vertex x_i to the color ξ whenever e_i is entered at time t_i^ξ within the exploration of (S_p, τ). We claim that c is a proper coloring. To see this, suppose on the contrary that there is a monochromatic edge $x_j x_k$ with $j < k$ of color ξ in G. Then, this means that e_j was entered at time t_j^ξ and exited at time at least $t_j^{\xi+1}$ and similarly e_k was entered at time t_k^ξ and exited at time at least $t_k^{\xi+1}$. But then, since every time in $\tau(e_{jk}^\xi)$ is contained either in the open interval $(t_j^\xi, t_j^{\xi+1}) \subseteq \mathbb{R}$ or the open interval $(t_k^\xi, t_k^{\xi+1}) \subseteq \mathbb{R}$ we know that e_{jk}^ξ cannot be explored (by (4)). This contradicts the assumption that (S_p, τ) is explorable, hence c must be a proper coloring.

Conversely, suppose G admits a proper 3-coloring $c : V(G) \to \{0, 1, 2\}$**.** We define the following exploration of (S_p, τ) (see Fig. 1):

- for every vertex x_i in G, if $c(x_i) = \xi$, then visit e_i at $(t_i^\xi, t_i^{\xi+1})$
- for every edge $x_j x_k$ in G with $j < k$ and every color $\xi \in \{0, 1, 2\}$, define the visit of e_{jk}^ξ as follows: if $c(x_j) \neq \xi$, then visit e_{jk}^ξ at $(t_j^\xi + 2k - 1, t_j^\xi + 2k)$; otherwise visit e_{jk}^ξ at $(t_k^\xi + 2j - 1, t_k^\xi + 2j)$.

Our aim now is to show that there are no conflicts between the visits we have just defined in terms of the coloring c (and thus that they witness the explorability of (S_p, τ)).

Take any $i < j$ and any $\xi \in \{0, 1, 2\}$. By our definition of $\tau(e_i)$ and $\tau(e_j)$, we must have $\max \tau(e_i) = t_i^3 < 2jn^2 = \min \tau(e_j)$ whenever $i < j$. Thus, we note that there are no conflicts between the visit of e_i and the visit of e_j.

Note that, for all $(\xi, \omega) \in \{0, 1, 2\}^2$ and all pairs of edges $x_i x_j$ and $x_k x_\ell$ in G with $i < j$ and $k < \ell$, the visit $(v_{i,j}, v_{i,j} + 1)$ of e_{ij}^ξ is in conflict with the visit $(v_{k,\ell}, v_{k,\ell} + 1)$ of $e_{k\ell}^\omega$ only if $x_i x_j = x_k x_\ell$. To see this, recall that the first time in any potential visit of any edge e_{ij}^ξ was defined to be either

$$t_i^\xi + 2j - 1 = 2in^2 + 2\xi(n + 1) + 2k - 1 \text{ or}$$
$$t_j^\xi + 2i - 1 = 2jn^2 + 2\xi(n + 1) + 2i - 1$$

and both of these are odd (a property that does not depend on i and j). Thus, the visits of e_{ij}^ξ and $e_{k\ell}^\omega$ both consist of two consecutive times where, as we just observed, the first time is odd. This means that we would only have a conflict if $v_{i,j} = v_{k,\ell}$ which, as can be easily checked, happens only if $i = k$ and $j = \ell$.

Finally we claim that there are no conflicts between the visit of e_{ij}^ξ and the visits of either e_i or e_j. To show this, we will only argue for the lack of conflicts between the visits of e_i and e_{ij}^ξ since the same ideas suffice for the e_j-case as well. Suppose $c(x_i) = \xi$, then we visit e_{ij}^ξ at $(t_j^\xi + 2i - 1, t_j^\xi + 2i)$ and then

$$t_j^\xi + 2i - 1 > t_j^\xi > t_i^3 = \max \tau(e_i)$$

since $i < j$ and since c is a proper coloring. Similarly, if $c(x_i) \neq \xi$, then we visit e_{ij}^ξ at $(t_i^\xi + 2j - 1, t_i^\xi + 2j)$. As we observed in Inequality (4), we have $t_i^\xi < t_i^\xi + 2j - 1 <$

$t_i^{\xi} + 2j < t_i^{\xi+1}$. Thus, if (u_1, u_2) is the visit of e_i, then either $u_2 < t_i^{\xi}$ or $t_i^{\xi+1} < u_1$. In other words, no conflicts arise.

This concludes the proof since we have shown that the visits we assigned to the edges of S_p constitute an exploration of (S_p, τ). □

Observe that increasing the maximum number of times per edge cannot make the problem easier: we can easily extend the hardness result to any $k' > 4$ by simply adding a new edge with k' times all prior to the times that are already in the star (recall that the parameter k in StarExp(k) is the *maximum* number of appearances of any edge). This, together with the fact that Akrida, Mertzios and Spirakis [15] showed that StarExp(k) is in NP for all $k \geq 0$, allows us to conclude the following corollary.

Corollary 2 *For all k at least* 4, StarExp(k) *is* NP-*complete.*

Next we shall reduce StarExp(k) to TempEuler($k-1$). We point out that, for our purposes within this section, only the first point of the statement of Lemma 3 is needed. However, later (in the proof of Corollary 9) we shall make use of the properties stated in the second point of Lemma 3 (this is also why we allow any k times per edge rather than just considering the case $k = 4$); specifically, this second point captures the fact that the reduction only changes the interval-membership-width (the new parameter we define in Sect. 4) by a constant factor. Thus, we include full details here.

Lemma 3 *For all $k \geq 2$ there is a polynomial-time-computable mapping taking every* StarExp(k) *instance (S_n, τ) to a* TempEuler($k-1$) *instance (D_n, σ) such that*

1. *(S_n, τ) is a yes instance for* StarExp(k) *if and only if (D_n, σ) is a yes instance for* TempEuler($k-1$) *and*
2. *D_n is a graph obtained by identifying n-copies $\{K_1^3, \ldots, K_n^3\}$ of a* 3*-cycle along one center vertex (see Fig. 2) and such that*

$$\max_{t \in \mathbb{N}} |\{e \in E(D_n) : \min(\sigma(e)) \leq t \leq \max(\sigma(e))\}| \\ \leq 3 \max_{t \in \mathbb{N}} |\{e \in E(S_n) : \min(\tau(e)) \leq t \leq \max(\tau(e))\}|.$$

Proof Note that we can assume without loss of generality that all times of every edge in S_n are even (if not, then just double every time; this cannot change explorability) and that every time has at least two appearances (otherwise we trivially have a no-instance).

Now we show how to construct a TempEuler($k-1$)-instance (D_n, σ) from (S_n, τ) such that (D_n, σ) is temporally Eulerian if and only if (S_n, τ) is explorable (see Fig. 2). The idea is to construct D_n by adding a triangle C_e for each edge $e \in E(S_n)$ in such a way that visiting e in an exploration of (S_n, τ) corresponds to walking along the triangle C_e (imagine "stretching" every leaf of the star into an edge and then "spreading" the times of the edge appropriately along this new cycle). To state this precisely, denote the vertices of the i-th 3-cycle C_i^3 of D_n by $\{c, x_{i,1}, x_{i,2}\}$ and let its edges be $f_{i,1} = cx_{i,1}$, $f_{i,2} = x_{i,1}x_{i,2}$ and $f_{i,3} = x_{i,2}c$. For every $i \in [n]$ and $\ell \in [k]$ with $\tau(e_i) = \{t_1, \ldots, t_\ell\}$

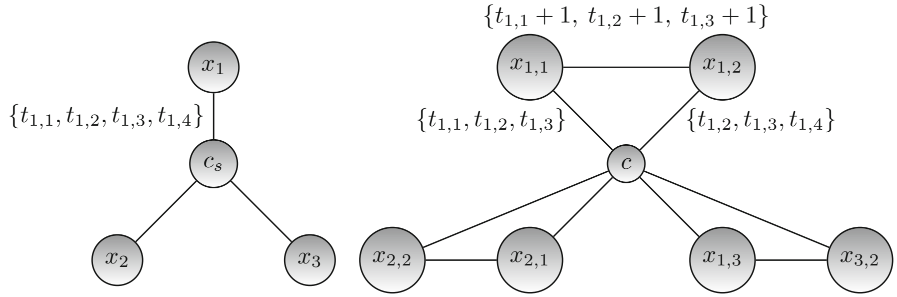
x_1
$\{t_{1,1}, t_{1,2}, t_{1,3}, t_{1,4}\}$
c_s
x_2
x_3
$\{t_{1,1}+1,\ t_{1,2}+1,\ t_{1,3}+1\}$
$x_{1,1}$
$x_{1,2}$
$\{t_{1,1}, t_{1,2}, t_{1,3}\}$
$\{t_{1,2}, t_{1,3}, t_{1,4}\}$
c
$x_{2,2}$
$x_{2,1}$
$x_{1,3}$
$x_{3,2}$

Fig. 2 Building (D_3, σ) from (S_3, τ). The times along edges are drawn only for the edge $c_s x_1$ in S_3 and for its corresponding 3-cycle $c x_{1,1} x_{1,2}$ in D_3. Since $t_{1,1}, t_{1,2}, t_{1,3}$ and $t_{1,4}$ are all multiples of 2, we know that $t_{1,j} < t_{1,j+1} < t_{1,j+1}$ for all $j \in [3]$. Thus, the reduction associates the visit (t_s, t_e) of $c_s x_1$ in the star to exploration $(t_s, t_s + 1, t_e)$ of the 3-cycle corresponding to $c_s x_1$ in D_3

where $t_1 < t_2 < \cdots < t_\ell$, define the map $\sigma : E(D_n) \to 2^{\mathbb{N}}$ as:

$$\begin{aligned}\sigma(f_{i,1}) &:= \{t_1, \ldots, t_{\ell-1}\},\\ \sigma(f_{i,2}) &:= \{t_1 + 1, \ldots, t_{\ell-1} + 1\},\\ \sigma(f_{i,3}) &:= \{t_2, \ldots, t_\ell\}.\end{aligned}$$

Note that $\sigma(f_{i,1})$, $\sigma(f_{i,2})$ and $\sigma(f_{i,3})$ all have size at most $k-1$. Now suppose (S_n, τ) is a yes-instance witnessed by the sequence $\mathcal{V}$ of visits $\mathcal{V} := (x_1, y_1), \ldots, (x_n, y_n)$ of the edges $e_1, \ldots, e_n$ of S_n and observe that $y_i < x_{i+1}$ for all $i \in [n-1]$. We claim that the sequence of time-edges

$$(f_{1,1}, x_1), (f_{1,2}, x_1 + 1), (f_{1,3}, y_1), \ldots, (f_{n,1}, x_n), (f_{n,2}, x_n + 1), (f_{n,3}, y_n)$$

is a temporal Eulerian circuit in G. To see this, recall that $y_j < x_{j+1}$ (for $j \in [n-1]$) and note that:

1. by definition $f_{1,1}, f_{1,2}, f_{1,3}, \ldots, f_{n,1}, f_{n,2}, f_{n,3}$ is an Eulerian circuit in the underlying static graph D_n (i.e. we walk along each 3-cycle in turn) and
2. $x_i < x_i + 1 < y_i$ for all $i \in [n]$ since we assumed that $x_i, y_i \in 2\mathbb{N}$.

Conversely, suppose (D_n, σ) is a yes-instance and let this fact be witnessed by the temporal Eulerian circuit $\mathcal{K}$. Recall that a temporal Eulerian circuit induces an Eulerian circuit in the underlying static graph. Thus, since every Eulerian circuit in D_n must run through each 3-cycle, we know that $\mathcal{K}$ must consist – up to relabelling of the edges – of a sequence of time-edges of the form

$$\begin{aligned}\mathcal{K} :=&(f_{1,1}, x_{1,1}), (f_{1,2}, x_{1,2}), (f_{1,3}, y_{1,3}), \quad (f_{2,1}, x_{2,1}), (f_{2,2}, x_{2,2}), (f_{2,3}, y_{2,3}),\\ &\ldots, \quad (f_{n,1}, x_{n,1}), (f_{n,2}, x_{n,2}), (f_{n,3}, y_{n,3}).\end{aligned}$$

It follows immediately from the definition of (D_n, σ) that visiting each edge e_j in S_n at $(x_{j,1}, y_{j,3})$ constitutes an exploration of (S_n, τ), as desired. □

Since TEMPEULER(k) is clearly in NP (where the circuit acts as a certificate), our desired NP-completeness result follows immediately from Lemma 3 and Corollary 2.

Corollary 4 TEMPEULER(k) *is* NP-*complete for all k at least* 3.

As we noted earlier, TEMPEULER(1) is trivially solvable in time linear in the number of edges of the underlying static graph. Although our proof leaves open the $k = 2$ case, we note that this was resolved by Marino and Silva [19] since the dissemination of an earlier version of this paper: they demonstrated that TEMPEULER(k) is NP-complete for all $k \geq 2$.

Observe that the reduction in Lemma 3 rules out FPT algorithms with respect to many standard parameters describing the structure of the underlying graph (for instance the path-width is 2 and feedback vertex number[2] is 1). In fact we can strengthen these intractability results even further by showing that TEMPEULER(k) is hard even for instances whose underlying static graph has vertex-cover number[3] 2 (Theorem 5 below). This motivates our search in Sect. 4 for parameters that describe the structure of the times assigned to edges rather than just the underlying static structure.

Theorem 5 *For all $k \geq 4$, the* TEMPEULER(k) *problem is* NP-*complete even on temporal graphs whose underlying static graph has vertex-cover number* 2.

Proof Take any STAREXP(k) instance (S_n, τ) and assume that n is even (if not, then simply add a dummy edge with all appearances strictly after the lifetime of the graph). Denoting by c the center of S_n and by $x_1, \ldots, x_n$ its leaves, let $S_n^{c_1,c_2}$ be the double star constructed from S_n by splitting c into two twin centers; stating this formally, we define $S_n^{c_1,c_2}$ as

$$S_n^{c_1,c_2} = (\{c_1, c_2, x_1, \ldots, x_n\}, \{c_i x_j : i \in [2] \text{ and } j \in [n]\}).$$

Notice that, since n is even, $S_n^{c_1,c_2}$ is Eulerian and notice that the set $\{c_1, c_2\}$ is a vertex cover of $S_n^{c_1,c_2}$.

Defining $\sigma : E(S_n^{c_1,c_2}) \to 2^{\mathbb{N}}$ for all $i \in [2]$ and $j \in [n]$ as $\sigma(c_i x_j) = \tau(c x_j)$, we claim that the temporal graph $(S_n^{c_1,c_2}, \sigma)$ is temporally Eulerian if and only if (S_n, τ) is explorable.

Suppose that (S_n, τ) is explorable and let this be witnessed by the sequence of visits $(s_1, t_1), \ldots, (s_n, t_n)$. Then it follows immediately by the definition of σ that the following sequence of time-edges is a temporal circuit in $(S_n^{c_1,c_2}, \sigma)$:

$$(c_1 x_1, s_1), (x_1 c_2, t_1), (c_2 x_2, s_2)(x_2 c_1, t_2), (c_1 x_3, s_3), (x_3 c_2, t_3), \\ \ldots, (c_2 x_n, s_n)(x_n c_1, t_n).$$

[2] Recall that a *feedback vertex set* in a graph (resp. directed graph) G is a vertex subset $S \subseteq V(G)$ such that $G - S$ is a forest (resp. directed acyclic graph). The *feedback vertex number* of a graph G is the minimum number of vertices needed to form a feedback vertex set in G.

[3] Recall that a *vertex-cover* in a graph G is a vertex-subset $S \subseteq V(G)$ such that every edge in G is incident with at least one vertex in S. The *vertex-cover number* of a graph G is the minimum number of vertices needed to form a vertex-cover of G.

To see this, note that this clearly induces an Eulerian circuit in the underlying static graph $S_n^{c_1,c_2}$; furthermore, since $(s_1, t_1), \ldots, (s_n, t_n)$ is an exploration in (S_n, τ), it follows that $s_1 < t_1 < s_2 < t_2 < \ldots < s_n < t_n$, as desired.

Suppose now that $(S_n^{c_1,c_2}, \sigma)$ is temporally Eulerian and that this fact is witnessed (without loss of generality – up to relabeling of vertices) by the temporal Eulerian circuit

$$(c_1x_1, s_1), (x_1c_2, t_1), (c_2x_2, s_2)(x_2c_1, t_2), (c_1x_3, s_3), (x_3c_2, t_3), \\ \ldots, (c_2x_n, s_n)(x_nc_1, t_n).$$

Then, by the definition of σ in terms of τ and by similar arguments to the previous case, we have that $(s_1, t_1), \ldots, (s_n, t_n)$ is an exploration of (S_n, τ). □

Notice that the reduction in the proof of Theorem 5 is from StarExp(k) (rather than from StarExp($k + 1$) as in Lemma 3) to TempEuler(k), so, in contrast to the reduction in Lemma 3, the proof of Theorem 5 cannot be used to show hardness of TempEuler(3).

4 Interval-Membership-Width

As we saw in the previous section, both TempEuler(k) and StarExp($k + 1$) are NP-complete for all $k \geq 3$ even on instances whose underlying static graphs are both sparse and highly structured (for instance even on graphs with vertex cover number 2). This suggests that any useful parameterization must take into account the *temporal structure* of the input. As we discussed previously, other authors have already proposed measures of this kind such as the temporal feedback vertex number [10] or temporal analogues of tree-width [21]. However these measures are all bounded on temporal graphs for which the underlying static graph has bounded feedback vertex number and tree-width respectively. Our reductions therefore show that TempEuler(k) is para-NP-complete with respect to these parameters. Thus, we do indeed need some new measure of temporal structure. To that end, here we introduce a parameter called *interval-membership-width* which depends only on temporal structure and not on any structural properties of the underlying graph other than the number of edges.[4] Parameterizing by this measure, we show that both TempEuler(k) and StarExp(k) lie in FPT.

We begin by reiterating the intuition behind our width measure. Recall from the discussion in the introduction that TempEuler(1) can be solved in time linear in $|E(G)|$. Moreover, the same is true for any TempEuler(k)-instance (G, τ) in which every edge is assigned a "private" interval of times: that is to say that, for all distinct edges e and f in G, either $\max \tau(f) < \min \tau(e)$ or $\max \tau(e) < \min \tau(f)$. This holds because, on instances of this kind, there is only one possible relative ordering of edges available for an edge-exploration. It is thus natural to expect that, for graphs whose edges have intervals that are "almost private" (defined formally below), we should be able to deduce similar tractability results.

[4] If (G_1, τ_1) and (G_2, τ_2) are two temporal graphs and there is a bijection $\pi : E(G_1) \to E(G_2)$ such that $\tau_1(e) = \tau_2(\pi(e))$ for all $e \in E(G_1)$, then (G_1, τ_1) and (G_2, τ_2) have the same interval-membership-width.

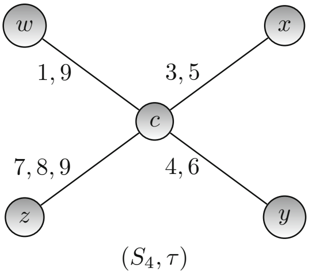

Fig. 3 A temporal star (S_4, τ) with interval-membership sequence: $F_1 = F_2 = \{cw\}$, $F_3 = \{cw, cx\}$, $F_4 = F_5 = \{cw, cx, cy\}$, $F_6 = \{cw, cy\}$ and $F_7 = F_8 = F_9 = \{cw, cz\}$

Towards a formalization of this intuition, consider a temporal graph (G, τ) as before, except that two of its edges e and f have $|\tau(e)| = |\tau(f)| = 2$ such that there is a time t with $\min \tau(e) \leq t \leq \max \tau(e)$ and $\min \tau(f) \leq t \leq \max \tau(f)$. It is easy to see that the TEMPEULER(k) problem is still tractable on graphs such as (G, τ) since there are only two possible relative edge-orderings for an edge exploration of (G, τ) (depending on whether we choose to explore e before f or f before e). These observations lead to the following definition of *interval-membership-width* of a temporal graph (see Fig. 3).

Definition 2 The *interval-membership sequence* of a temporal graph (G, τ) is the sequence $(F_t)_{t \in [\Lambda]}$ of edge-subsets of G where $F_t := \{e \in E(G) : \min \tau(e) \leq t \leq \max \tau(e)\}$ and Λ is the lifetime of (G, τ). The *interval-membership-width* of (G, τ) is the integer $\mathbf{imw}(G, \tau) := \max_{t \in \mathbb{N}} |F_t|$.

Note that a temporal graph has unit interval-membership-width if and only if every edge is active at times spanning a "private interval". Furthermore, we point out that the interval-membership sequence of a temporal graph is not the same as the sequence $(E_t(G, \tau))_{t \in \mathbb{N}}$. In fact, although $\max_{t \in \mathbb{N}} |E_t(G, \tau)| \leq \mathbf{imw}(G, \tau)$, there exist classes $\mathcal{C}$ of temporal graphs with unbounded interval-membership-width but such that every temporal graph in $\mathcal{C}$ satisfies the property that at most one edge is active at any given time. To see this consider any graph H with edges $e_1, \ldots, e_m$ and let (H, ν) be the temporal graph defined by $\nu(e_i) := \{i, m + i\}$. Clearly $\max_{i \in \mathbb{N}} |E_i(H, \nu)| = 1$, but we have $\mathbf{imw}(H, \nu) = m$.

Before continuing, we first note how to compute the interval-membership sequence of a temporal graph (G, τ) with lifetime Λ in $\mathcal{O}\big(\mathbf{imw}(G, \tau)\Lambda)\big)$ time. We point out now that, since this running time is linear in Λ, all of the running times in all of the following algorithmic results parameterized by interval-membership-width also include the time needed to compute the interval-membership sequence.

Lemma 6 *There is an algorithm that, given a temporal graph (G, τ) with every edge active at-least once and with lifetime Λ, computes the interval-membership sequence of (G, τ) in time $\mathcal{O}(w\Lambda)$ where $w = \mathbf{imw}(G, \tau)$.*

Proof Supposing $E(G) = \{e_1, \ldots e_m\}$, consider the following algorithm:

- initialize a list $(F_t)_{t \in [\Lambda]}$ with $F_t = \emptyset$ for all t,
- for each edge $e \in E(G)$,

S1 compute $m_e = \min \tau(e)$ and $M_e = \max \tau(e)$

S2 add e to every set F_i with $m_e \leq i \leq M_e$.

For each edge e, Step **S1** takes $\mathcal{O}(|\tau(e)|)$ time while Step **S2** takes $\mathcal{O}(|\{t : e \in F_t\}|)$ time. Thus, since $\tau(e) \subseteq \{t : e \in F_t\}$, steps **S1** and **S2** together take time $\mathcal{O}(|\{t : e \in F_t\}|)$. Summing over all edges of G, we have that, since every edge is active at-least once, the whole algorithm runs in time

$$\mathcal{O}\Big(\sum_{e \in E(G)} |\{t : e \in F_t\}|\Big) = \mathcal{O}\Big(\sum_{t \in [\Lambda]} |F_t|\Big) = \mathcal{O}(w\Lambda).$$

□

Armed with the notion of interval-membership-width, we now show that both TEMPEULER(k) and STAREXP(k) are in FPT when parameterized by this measure. We do so first for TEMPEULER(k) (Theorem 7) and then we leverage the reduction of Lemma 3 to deduce the fixed-parameter-tractability of STAREXP(k) as well (Corollary 9).

Theorem 7 *There is an algorithm that, given any temporal graph* (G, τ) *with lifetime* Λ, *decides whether* (G, τ) *is a yes-instance of* TEMPEULER(k) *in time* $\mathcal{O}(w^3 2^w \Lambda)$ *where* $w = \mathbf{imw}(G, \tau)$ *is the interval-membership-width of* (G, τ).

Proof Let $(F_t)_{t \in [\Lambda]}$ be the interval-membership sequence of (G, τ) and suppose without loss of generality that F_1 is not empty.

We now describe an algorithm that proceeds by dynamic programming over the sequence $(F_i)_{i \in [\Lambda]}$ to determine whether (G, τ) is temporally Eulerian. For each set F_i we compute a set $L_i \subseteq F_i^{\{0,1\}} \times V(G) \times V(G)$ consisting of triples of the form (f, s, x) where s and x are vertices in G and f is a function mapping each edge in F_i to an element of $\{0, 1\}$. Intuitively each entry (f, s, x) of L_i corresponds to the existence of a temporal walk starting at s and ending at x at time at most i and such that, for any edge $e \in F_i$, we will have $f(e) = 1$ if and only if e was traversed during this walk.

We now define the entries L_i recursively starting from the dummy set $L_0 := \{(\mathbf{0}, x, x) : \exists e \in F_1 \text{ incident with } x\}$ where $\mathbf{0} : e \in F_1 \mapsto 0$ is the function mapping every element in F_1 to 0 (here, in a slight abuse of notation, we are adopting the convention that there is always, at any time a zero-edge temporal walk from any vertex x to itself). Take any (f, s, y) in $F_i^{\{0,1\}} \times V(G) \times V(G)$. For (f, s, y) to be in L_i we require there to be an entry (g, s, x) of L_{i-1} such that

$$g(e) = 1 \text{ for all } e \in F_{i-1} \setminus F_i \tag{5}$$

and such that one of the following holds:

C1 $y = x$ and $f(e) = 1$ if and only if $e \in F_{i-1} \cap F_i$ and $g(e) = 1$,
or

C2 there exists an edge xy in G such that:

C2.P1 $xy \in E_i(G, \tau) \setminus \{e \in F_i : g(e) = 1\}$ and

C2.P2 $f(e) = 1$ if and only if $g(e) = 1$ or $e = xy$.

The Cases **C1** and **C2** correspond to the the two available choices we have when extending a temporal (s, x)-walk at time i: either we stay put at x (Case **C1**) or we find some new edge xy active at time i (Case **C2**) which has never been used before (Property **C2.P1**) and add it to the walk (Property **C2.P2**). Equation (5) ensures that we filter out partial solutions that we already know cannot be extended to a Eulerian circuit. To see this, note that, if an edge e will never appear again after time $i - 1$ and we have $g(e) = 0$, then there is no way of extending the temporal walk represented by the triple (g, s, x) to an Eulerian circuit in (G, τ) because one edge will always be left out (namely the edge e).

We claim that the input (G, τ) is temporally Eulerian if and only if L_Λ contains an entry $(\mathbf{1}, s, x)$ with $s = x$ and such that $\mathbf{1} : y \in F_\Lambda \mapsto 1$ is the constant all-1 function. To show this, we prove the following stronger claim.

Claim 8 *For all $i \in [\Lambda]$, L_i contains an entry (f, s, x) if and only if there exists a temporal walk $(e_1, t_1) \dots (e_p, t_p)$ starting at s and ending at x with $t_p \leq i$ and in which no edge is repeated and such that:*

IH1 $(F_1 \cup \dots \cup F_{i-1}) \setminus F_i \subseteq \{e_1, \dots, e_{p-1}\}$ *(i.e. every edge whose last appearance is before time i is traversed by the walk) and*

IH2 *for all $e \in F_i$, we have $f(e) = 1$ if $e \in \{e_1, \dots, e_p\}$ and $f(e) = 0$ otherwise (i.e. f correctly records which edges in F_i have been used in a walk).*

Proof of Claim 8 We show this by induction on i. The Claim holds trivially for $i = 0$, so suppose now that we are at some time $i > 0$ and hypothesise that the Claim holds for time $i - 1$. Furthermore denote by $W_i(f, s, x)$ the set of all temporal (s, x)-walks $(e_1, t_1) \dots (e_p, t_p)$ with $t_p \leq i$ which satisfy Properties **IH1** and **IH2**.

($\Longrightarrow$) First we show that if (f, s, y) is in L_i, then $W_i(f, s, x)$ is non-empty. By the construction of L_i, we know that, for (f, s, y) to be in L_i, there must have been an element (g, s, x) of L_{i-1} satisfying Eq. (5) from which we built (f, s, y) according to either Case **C1** or Case **C2**.

Suppose we applied Case **C1** to add (f, s, y) to L_i (i.e. we "extended" some walk in $W_{i-1}(g, s, y)$ by deciding not to move). Then we know that $f(e) = 1$ if and only if $g(e) = 1$. Notice that any walk corresponding to (f, s, y) cannot fail to visit some edge in $E_{i-1}(G, \tau)$ that will never again be active after time $i-1$ since we know that (g, s, y) satisfies Eq. (5). In particular, there is a walk W in $W_{i-1}(f, s, y) \subseteq W_i(f, s, y)$ satisfying Properties **IH1**. Furthermore, f satisfies Property **IH2** since g satisfies it by induction and $f = g$. Thus, we have shown that, if we applied Case **C1** to add (f, s, y) to L_i, then $W_i(f, s, y) \neq \emptyset$.

Suppose instead that we applied Case **C2** to add (f, s, y) to L_i. In other words suppose we found an edge xy active at time i with which we wish to extend some walk $W := (e_1, t_1), \dots, (e_p, t_p)$ in $W_{i-1}(g, s, x)$ which starts at s and ends at x. Note that we can infer that $W' := (e_1, t_1) \dots (e_p, t_p)(xy, i)$ is a valid temporal (s, y)-walk with no repeated edges since:

- W has no repeated edges (by the induction hypothesis),
- xy was not traversed by W (by Property **C2.P1**) and

- $t_p \leq i-1$ (since $W \in W_{i-1}(g,s,x)$).

Thus, the fact that g satisfies equation (5) combined with the induction hypothesis implies that every edge whose last appearance is before time i is traversed by W' (i.e. W' satisfies Property **IH1**). Furthermore f satisfies Property **IH2** since g does and since $f(e)=1$ if and only if $g(e)=1$ or $e=xy$ (by Property **C2.P2**). Thus, we have shown that, if $(f,s,y) \in L_i$, then $W_i(f,s,y) \neq \emptyset$.

($\Longleftarrow$) Conversely, we now show that, if $W_i(f,s,y)$ is non-empty, then $(f,s,y) \in L_i$. Let W' be an element of $W_i(f,s,y)$ and let (xy,j) be the last time-edge traversed by W' (note $j \leq i$).

If $j < i$ then, by the induction hypothesis, we get that $(f,s,y) \in L_{i-1}$ with $W' \in W_{i-1}(f,s,y)$. But then by the construction of L_i from L_{i-1} we have that $(f,s,y) \in L_i$ as desired.

Thus, suppose $j=i$. Then $W'-(xy,j)$ is a temporal (s,x)-walk ending at time at most $i-1$ satisfying Property **IH1**. Furthermore, by the induction hypothesis, there must be a $(g,s,x) \in L_{i-1}$ which satisfies Eq. (5) and such that $W'-(xy,j) \in W_{i-1}(g,s,x)$. Now note that, since (f,s,y) satisfies Properties **IH1** and **IH2**, we have that Properties **C2.P1** and **C2.P2** hold as well: thus, $(f,s,y) \in L_i$. □

Finally we consider the running time of the algorithm. First of all notice that we can compute L_{i+1} from L_i in time at most

$$(|E_{i+1}(G,\tau)|+1) \cdot |L_i| \leq (|F_{i+1}|+1) \cdot |L_i| \leq (w+1) \cdot |L_i|$$

(where the last inequality follows since $|F_i| \leq \mathbf{imw}(G,\tau)=w$ for any i). To see this, note that we construct the elements of L_{i+1} by iterating through the elements of L_i and considering for each one the $|E_{i+1}(G,\tau)|+1$ ways of taking a next step in a temporal walk at time i. Since we perform this computation Λ times (once for each time $i \in [\Lambda]$ where Λ is the lifetime), the whole algorithm runs in time $\mathcal{O}(w\Lambda \max_{i \in \Lambda}|L_i|)$. Thus, all that remains to be shown to complete the proof is that $|L_i|$ is $\mathcal{O}(w^2 2^w)$ for all i. Note that, from its definition, we already know that L_i has cardinality at most $\mathcal{O}(2^w n^2)$ since $L_i \subseteq F_i^{\{0,1\}} \times V(G) \times V(G)$. To improve this bound, we show that the following two statements hold:

RT1 there exists a time t such that every temporal Eulerian circuit in (G,τ) must start with a vertex incident with an edge in the bag F_t of the interval-membership sequence of (G,τ);

RT2 there exists a constant γ such that, for all i, let $\mathcal{X}_i \subseteq V(G)$ be the set of vertices of G defined as

$$\mathcal{X}_i := \{x \in V(G) : (f,s,x) \in L_i\}; \tag{6}$$

then $\mathcal{X}_i$ has cardinality at most γw, where $w = \mathbf{imw}(G,\tau)$.

To see why it suffices to prove claims **RT1** and **RT2**, notice that they imply that we not only have $L_i \subseteq F_i^{\{0,1\}} \times V(G) \times V(G)$ (which was how we defined L_i in the first place) but in fact there must always exist a $t \in [\Lambda]$ and a subset $\mathcal{X}_i \subseteq V(G)$ (for all

i) such that L_i is always of the form

$$L_i \subseteq F_i^{\{0,1\}} \times V(F_t) \times \mathcal{X}_i$$

where both $|V(F_t)|$ and $|\mathcal{X}_i|$ are $\mathcal{O}(w)$. This would clearly then imply that $|L_i|$ is $\mathcal{O}(w^2 2^w)$ for all i, as desired.

Proof of Claim RT1 Choose $t \in \mathbb{N}$ be greatest possible such that $\bigcup_{j \in [t]} F_j \subseteq F_t$. Suppose by way of contradiction that there exists a temporal Eulerian circuit that starts at a vertex s with s *not* incident with any edge in F_t. Let t' be the earliest time such that the bag $F_{t'}$ contains an edge which which s is incident.

Notice that, since t was chosen greatest possible such that $\bigcup_{j \in [t]} F_j \subseteq F_t$ and since s is not incident with any edge in F_t, it follows that $t' > t$ and that there exists an edge $e \in F_t \setminus F_{t'}$. But then we have a contradiction since $\max(\tau(e)) \leq t < t'$ and, by time t', e has not yet been visited by the temporal Eulerian circuit starting at s (i.e. any such circuit never visits the edge e). □

Proof of Claim RT2 We show that $\mathcal{X}_i$ has cardinality at most $6w$. Seeking a contradiction, suppose $|\mathcal{X}_i| \geq 6w + 1$. Since $|F_i| \leq w$, the set

$$\mathcal{X}_{\notin i} = \{x \in \mathcal{X}_i : \forall e \in F_i, e \text{ not incident with } z\}$$

of elements of $\mathcal{X}_i$ that are not incident with any edge in F_i consists of at least $4w + 1$ vertices.

Let $\xi : \mathcal{X}_{\notin i} \to E(G)$ be the map associating to each vertex z in $\mathcal{X}_{\notin i}$ an edge $\xi(z)$ such that: (1) $\xi(z)$ is incident with z, (2) $\min \tau(\xi(z)) \leq i$ and (3) the last appearance of $\xi(z)$ is latest possible. To see that ξ is well-defined, notice that by the definition of $\mathcal{X}_i$ (Eq. (6)) and since $\mathcal{X}_{\notin i} \subseteq \mathcal{X}_i$ we must have that, for every $z \in \mathcal{X}_{\notin i}$, there is a temporal walk ending at z at some time at most i which implies that the set

$$\{e \in E(G) : e \text{ incident with } z \text{ and } \min \tau(e) \leq i\}$$

must be non-empty. Furthermore, notice that $\max \tau(\xi(z)) < i$ since otherwise, by the definition of $\mathcal{X}_i$, we would have that $\xi(z) \in F_i$ (which contradicts the fact that $z \in \mathcal{X}_{\notin i}$).

Pick a vertex $z \in \mathcal{X}_{\notin i}$ such that $\max \tau(\xi(z)) \leq \max \tau(\xi(z'))$ for any other $z' \in \mathcal{X}_{\notin i}$. Since $\mathcal{X}_{\notin i}$ contained at least $4w + 1$ elements, then $\xi(\mathcal{X}_{\notin i})$ consists of at least $2w + 1$ edges. This, together with the fact that $|F_{\max \tau(\xi(z))} \cup F_i| \leq 2w$, implies that there must an edge $\xi(y)$ with

$$\xi(y) \in \xi(\mathcal{X}_{\notin i}) \setminus (F_{\max \tau(\xi(z))} \cup F_i)$$

which implies that

$$\max \tau(\xi(z)) < \min \tau(\xi(y)) < \max \tau(\xi(y)) < i.$$

By the definition of $\mathcal{X}_i$ and since $z \in \mathcal{X}_i$, there is some $(f, s, z) \in L_i$ and, by the previous Claim, there is a walk $W \in W_i(f, s, z)$ satisfying **IH1** and **IH2**. Notice that, since W ends at the vertex z, it must be that the last time we "took a step" on W was at a time at most $\max \tau(\xi(z))$; in particular, this means that we did not move from z at time i. But then, since $\max \tau(\xi(y)) < i$, y never appears again after time $i - 1$ and hence W never traverses $\xi(y)$: this contradicts Property **IH1**. □

As a corollary of Theorem 7, we can leverage the reduction of Lemma 3 to deduce that StarExp(k) is in FPT parameterized by the interval-membership-width.

Corollary 9 *There is an algorithm that, given a* StarExp(k) *instance* (S_n, τ)*, decides whether* (S_n, τ) *is explorable in time* $\mathcal{O}(w^3 2^{3w} \Lambda)$ *where* $w = \mathbf{imw}(S_n, \tau)$ *and* Λ *is the lifetime of the input.*

Proof By Lemma 3, we know that there is a polynomial-time reduction that maps any StarExp(k) instance (S_n, τ) to a TempEuler($k - 1$)-instance (D_n, σ) such that

$$\begin{aligned}&\max_t |\{e \in E(D_n) : \min(\sigma(e)) \leq t \leq \max(\sigma(e))\}| \\ &\quad \leq 3 \max_t |\{e \in E(S_n) : \min(\tau(e)) \leq t \leq \max(\tau(e))\}|.\end{aligned}$$

In particular, this implies that $\mathbf{imw}(D_n, \sigma) \leq 3w$. Thus, we can decide whether (S_n, τ) is explorable in time $\mathcal{O}(w^3 2^{3w} \Lambda)$ by applying the algorithm of Theorem 7 to (D_n, σ). □

5 Win-Win Approach to Regularly Spaced Times

In this section we find necessary conditions for edge-explorability of temporal graphs with respect to their interval-membership-width. This allows us to conclude that either we are given a no-instance or that the interval-membership-width is small (in which case we can employ our algorithmic results from the previous section).

We will apply this bidimensional approach to variants of TempEuler(k) and StarExp(k) in which we are given upper and lower bounds (u and ℓ respectively) on the difference between any two consecutive times at any edge (meaning that for any edge e active at times $t_1 < t_2 < \cdots < t_{|\tau(e)|}$, we have $\ell \leq t_{i+1} - t_i \leq u$ for all $1 \leq i < |\tau(e)|$). Specifically we show that StarExp(k) is in FPT parameterized by k, ℓ and u (Theorem 12) and that TempEuler(k) is in FPT parameterized by k and u (Theorem 13). In other words, these results allow us to trade in the dependences on the interval-membership-width of Corollary 9 and Theorem 7 for dependences on k, ℓ, u and k, u respectively.

We note that, for StarExp instances, the closer ℓ and u get, the more restricted the structure becomes to the point that the dependence on ℓ and u in the running time of our algorithm vanishes when $\ell = u$. In particular, this shows that the problem of determining the explorability of StarExp(k)-instances for which consecutive times at each edge are exactly λ time-steps apart (for some $\lambda \in \mathbb{N}$) is in FPT parameterized solely by k (Corollary 14). This partially resolves an open problem of Akrida, Mertzios and Spirakis [15] which asked to determine the complexity of exploring StarExp(k)-instances with evenly spaced times.

Towards these results, we first provide sufficient conditions for non-explorability of any StarExp(k) instance (Lemma 10). These conditions depend only on: (1) knowledge of the maximum and minimum differences between any two successive appearances of any edge, (2) the interval-membership-width and (3) the maximum number of appearances k of any edge.

Lemma 10 *Let (S_n, τ) be a temporal star with at most k times at any edge and such that every two consecutive times at any edge differ at least by ℓ and at most by u. If (S_n, τ) is explorable, then* $\mathbf{imw}(S_n, \tau) \leq (2(k-1)u + 1)/(\ell + 1)$.

Proof Let Λ be the lifetime of (S_n, τ), let $(F_t)_{t \in [\Lambda]}$ be the interval-membership sequence of (S_n, τ) and choose any $n \in [\Lambda]$ such that $|F_n| = \mathbf{imw}(S_n, \tau)$. Let m and M be respectively the earliest and latest times at which there are edges in F_n which are active and choose representatives e_m and e_M in F_n such that $m = \min \tau(e_m)$ and $M = \max \tau(e_M)$.

Recall that visiting any edge e in S_n requires us to pick two appearances (which differ by at least $\ell + 1$ time-steps) of e: one appearance is used to go along e from the center of S_n to the leaf and another appearance is used to return to the center of the star. Thus, whenever we specify how to visit an edge e of F_n, we remove at least $\ell + 1$ time-steps from the available time-set $\{m, \dots, M\}$ at which any other edge in F_n can be visited. Furthermore, since any exploration of (S_n, τ) must explore all of the edges in F_n, for (S_n, τ) to be explorable, we must have $|F_n|(\ell + 1) \leq 1 + M - m$. Thus, by the choice of F_n, we find that

$$\begin{aligned}
\mathbf{imw}(S_n, \tau)(\ell + 1) &= |F_n|(\ell + 1) \\
&\leq 1 + M - m \\
&\leq 1 + |\max \tau(e_M) - \min \tau(e_M)| + |\max \tau(e_m) - \min \tau(e_m)| \\
&\quad \text{(since, } \forall e \in F_n, \min \tau(e) \leq n \leq \max \tau(e)) \\
&\leq 2(k-1)u + 1
\end{aligned}$$

as desired. □

Notice that nearly-identical arguments yield the following slightly weaker result with respect to the TempEuler(k) problem.

Lemma 11 *Let (G, τ) be a* TempEuler(k) *instance such that every two consecutive times at any edge differ at most by u. If (G, τ) is temporally Eulerian, then* $\mathbf{imw}(G, \tau) \leq 2(k-1)u + 1$.

The reason that we can only bound $\mathbf{imw}(G, \tau)$ above by $2(k-1)u + 1$ (rather than $(2(k-1)u + 1)/(\ell + 1)$ as in the StarExp(k) case of Lemma 10) is that temporal Euler circuits only visit each edge once (so exploring each edge only removes exactly one available time).

Lemma 10 allows us to employ a "win-win" approach for StarExp(k) when we know the maximum difference between consecutive times at any edge: either the considered instance does not satisfy the conditions of Lemma 10 (in which case we have a no-instance) or the interval-membership-width is small enough for us to usefully apply Corollary 9. These ideas allow us to conclude the following result.

Theorem 12 *Let (S_n, τ) be a temporal star with at most k times at any edge and such that every two consecutive times at any edge differ at least by ℓ and at most by u. There is an algorithm that decides whether (S_n, τ) is explorable in time $2^{\mathcal{O}(ku/\ell)}\Lambda$ where Λ is the lifetime of the input.*

Proof Determine $\mathbf{imw}(S_n, \tau)$ (using the algorithm of Lemma 16); if $\mathbf{imw}(S_n, \tau) > (2(k-1)u+1)/(\ell+1)$, then (S_n, τ) is not explorable by Lemma 10. Otherwise run the algorithm given in Corollary 9. In this case, since $w := \mathbf{imw}(S_n, \tau) \leq (2(k-1)u+1)/(\ell+1)$, we know that the algorithm of Corollary 9 will run on (S_n, τ) in time $2^{\mathcal{O}(ku/\ell)}\Lambda$. □

Once again arguing by bidimensionality (this time using Lemma 11 and Theorem 7) we can deduce the following fixed-parameter tractability result for TempEuler.

Theorem 13 *Let (G, τ) be a* TempEuler(k) *instance such that every two consecutive times at any edge differ at most by u. There is an algorithm that decides whether (G, τ) is temporally Eulerian in time $2^{\mathcal{O}(ku)}\Lambda$ where Λ is the lifetime of the input.*

As a special case of Theorem 12 (i.e. the case where $\ell = u$) we partially resolve an open problem of Akrida, Mertzios and Spirakis [15] which asked to determine the complexity of exploring StarExp(k)-instances with evenly spaced times. In particular, we show that the problem of deciding the explorability of such evenly spaced StarExp(k)-instances is in FPT when parameterized by k.

Corollary 14 *There is an algorithm that, given any* StarExp(k) *instance (S_n, τ) with lifetime Λ such that every two pairs of consecutive times assigned to any edge in (S_n, τ) differ by the same amount, decides whether (S_n, τ) is explorable in time $2^{\mathcal{O}(k)}\Lambda$.*

6 A Vertex Version of Interval-Membership-Width

It is natural to ask whether the parameter interval-membership-width also allows the design of FPT algorithms for problems involving vertex exploration or reachablity. Unfortunately, it seems that this parameter is not so well-suited to problems in which we need to keep track of the set of vertices that have been reached or visited at a particular time in the evolution of the temporal graph. To illustrate this, we consider a problem MinReachDelete (defined formally below), which remains intractable even on graphs with interval-membership-width one. This motivates the introduction of a vertex variant of our parameter.

A vertex u is said to be *temporally reachable* from v in the temporal graph (G, τ) if there exists a temporal path from v to u; every vertex is assumed to be temporally reachable from itself. The *temporal reachability set* of vertex v in (G, τ), written $\mathbf{reach}_{G,\tau}(v)$, is then defined to be the set of vertices which are temporally reachable from v; the temporal reachability set of a set $S \subseteq V(G)$, written $\mathbf{reach}_{G,\tau}(S)$, is defined in the natural way to be $\bigcup_{v \in S} \mathbf{reach}_{G,\tau}(v)$. The *temporal reachability* of a set of vertices $S \subseteq V(G)$ is $|\mathbf{reach}_{G,\tau}(S)|$ (for a vertex v in (G,τ) we write $|\mathbf{reach}_{G,\tau}(v)|$ rather than $|\mathbf{reach}_{G,\tau}(\{v\})|$). We consider the following problem, introduced in [28].

MinReachDelete
Input: A temporal graph (G,τ), a set of source vertices S, and $k, h \in \mathbb{N}$.
Question: Is there a set E' of time-edges, with $|E'| \leq k$, such that the temporal reachability of S in $(G,\tau) \setminus E'$ is at most h?

Note that this is a generalisation of the problem TR- EdgeDeletion introduced in [29], where the set S of sources is always taken to be equal to $V(G)$. This problem was originally motivated by a practical application related to the design of optimal interventions to restrict the spread of an infectious disease through a contact network [29]. Here we adapt one of the arguments used to demonstrate intractability of TR Edge Deletion [29, Theorem 3.1] to show that MinReachDelete is para-NP-hard with respect to the interval-membership-width of the input graph.

Theorem 15 MinReachDelete *is NP-hard, even if the input temporal graph has interval-membership-width one.*

Proof We prove that the problem remains NP-hard even when the input temporal graph satisfies the following two properties:

1. every edge is active at exactly one time, and
2. no two edges are active simultaneously.

These two conditions together immediately imply that the graph has interval-membership-width one. Hardness of TR Edge Deletion, and hence MinReachDelete, when every edge appears exactly once, was already demonstrated in [29]; however, in this construction, an unbounded number of edges is active at the same time (giving unbounded interval-membership-width). Here we adapt the construction so that no two edges are active at the same time. Note that, as each edge is active at exactly one time in our construction, we can use the terms edge and time-edge interchangeably in the proof.

As in [29, Theorem 3.1], the reduction is from the NP-hard problem Clique. Let (G,r) be an instance of Clique, where $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$. We will construct an instance $((H,\tau),k,h)$ of MinReachDelete which is a yes-instance if and only if (G,r) is a yes-instance for Clique. As in the proof of [29, Theorem 3.1], we assume without loss of generality that $m > r + \binom{r}{2}$.

We begin by defining H. The vertex set of H is $V(H) = \{s\} \cup V(G) \cup E(G) \cup W$, where $W := \{w_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m, v_i \in e_j\}$. The edge set is

$$\begin{aligned} E(H) = &\{sv_i : 1 \leq i \leq n\} \\ &\cup \{v_i w_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m, v_i \in e_j\} \\ &\cup \{w_{i,j} e_j : 1 \leq i \leq n, 1 \leq j \leq m, v_i \in e_j\} \\ &\cup \{s w_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m, v_i \in e_j\}. \end{aligned}$$

We complete the construction of the temporal graph (H, τ) by setting

$$\tau(e) = \begin{cases} i & \text{if } e = sv_i \text{ for some } 1 \leq i \leq n \\ n + 2j - 1 & \text{if } e = v_i w_{i,j} \text{ for some } 1 \leq i \leq n \text{ and } 1 \leq j \leq m \\ n + 2j & \text{if } e = w_{i,j} e_j \text{ for some } 1 \leq i \leq n \text{ and } 1 \leq j \leq m \\ n + 2m + j & \text{if } e = sw_{i,j} \text{ for some } 1 \leq i \leq n \text{ and } 1 \leq j \leq n. \end{cases}$$

It is immediate from the construction of (H, τ) that each edge is active at exactly one time, and that no two edges are active at the same time. Finally, we set $S = \{s\}$, $k = r$ and $h = 1 + (n - r) + 2m + (m - \binom{r}{2})$.

Suppose first that $U \subseteq V(G)$ is a clique of size r in G. Set $E' := \{sv : v \in U\}$ and write (H', τ') for the temporal graph obtained from (H, τ) by deleting all edges in E'. We argue that the temporal reachability of s in (H', τ') is at most h, implying that $((H, \tau), \{s\}, k, h)$ is a yes-instance for MinReachDelete. Note that s reaches every vertex in W along one-edge paths, but that no further vertices can be reached along paths starting with these edges as each such edge is active strictly later than any other edge incident with the endpoint in W. It follows that every vertex reached by s that does not belong to W must be reached via a an element of $V(G) \setminus U$. Thus, we deduce that

$$\mathbf{reach}_{H',\tau'}(s) \subseteq \{s\} \cup W \cup V(G) \setminus U \cup \bigcup_{v \in V(G) \setminus U} \{e \in E(G) : v \in e\}.$$

In particular, we see that s does not reach any vertex in U, or any element of $E(G)$ with both endpoints in U. Since U induces a clique in G, we see that s fails to reach at least $r + \binom{r}{2}$ vertices in (H', τ') and hence $|\mathbf{reach}_{H',\tau'}(s)| \leq 1 + n + m + 2m - r - \binom{r}{2} = h$, as required.

Conversely, suppose that there exists a set $E' \subseteq E(H)$ with $|E'| = r$ such that, if (H', τ') is the graph obtained from (H, τ) by deleting all edges in E', we have that the reach of s is (H', τ') is at most $|\mathbf{reach}_{H',\tau'}(s)| \leq h$.

Suppose first that $E' \cap \{sw : w \in W\} = \emptyset$; we argue that in this case s reaches at least $|V(H)| - |E'| - \binom{|E'|}{2}$ vertices in (H', τ') and that this lower bound can only be achieved if G contains a clique on r vertices.

We begin by arguing that we may assume without loss of generality that every element of E' is incident with s. Suppose first that $v_i w_{i,j} \in E'$ for some $1 \leq i \leq n$ and $1 \leq j \leq m$. The only vertices which are reached from s along a temporal path using this edge are e_j and elements of W which are necessarily in the reachability set of s since we are assuming we do not delete any edge of the form sw with $w \in W$. We can therefore replace $v_i w_{i,j}$ with sv_i in E' without increasing the number of vertices that are temporally reachable from s, since deleting E' will still destroy the temporal path from s to e_j via $v_i w_{i,j}$. Suppose now that $w_{i,j} e_j \in E'$ for some $1 \leq i \leq n$ and $1 \leq j \leq m$. Again, the only vertex outside W that is reachable from s along a temporal path that includes this edge is e_j, and as before we can destroy this temporal path by instead deleting sv_i. We therefore conclude that, provided that $E' \cap \{sw : w \in W\} = \emptyset$, it is possible to delete a subset of $\{sv_i : 1 \leq i \leq n\}$ of size $|E'|$ such that the reachability set of s is a subset of $\mathbf{reach}_{H',\tau'}(s)$. We therefore assume from now on that $E' \subseteq \{sv_i : 1 \leq i \leq n\}$.

Set $U \subseteq V(G)$ to be the set of vertices in $V(G)$ that are incident with an edge in E'. We claim that U induces a clique in G. To see this, note that s reaches all of $V(G) \setminus U$, all of W, and every vertex in $E(G)$ that does not have both endpoints in U. By assumption, we therefore have

$$\begin{aligned} h \geq |\mathbf{reach}_{H',\tau'}(v)| &= 1 + |V(G) \setminus U| + |W| + \left| \bigcup_{v \in V(G)\setminus U} \{e \in E(G) : v \in e\} \right| \\ &= 1 + (n-r) + 2m + m - \{xy \in E(G) : x, y \in U\}. \end{aligned}$$

By definition of h, it follows that $|\{xy \in E(G) : x, y \in U\}| \geq \binom{r}{2}$; this holds if and only if U induces a clique in G, in which case we have equality. It therefore follows, as claimed, that s reaches at least $|V(H)| - |E'| - \binom{|E'|}{2}$ vertices in (H', τ'), and that this lower bound can only be achieved if G contains a clique on r vertices.

To complete the proof, we argue that the remaining case $E' \cap \{sw : w \in W\} \neq \emptyset$ cannot occur. Suppose that $|E' \cap \{sw : w \in W\}| = t > 0$. Note that the deletion of an edge sw with $w \in W$ can at most remove w from the reachability set of s, since the edge sw is active strictly later than any other edge incident with w and so cannot be part of a longer temporal path starting at s. If we set $E'' = E' \setminus \{sw : w \in W\}$, this observation combined with the reasoning above tells us that

$$\begin{aligned} |\mathbf{reach}_{H',\tau'}(s)| &\geq |V(H)| - |E''| - \binom{|E''|}{2} - t \\ &= |V(H)| - (r-t) - \binom{|E''|}{2} - t \\ &> |V(H)| - r - \binom{r}{2} = h, \end{aligned}$$

contradicting our choice of E'. □

We now introduce the vertex version of interval-membership-width, which captures the maximum number of vertices incident both to an edge active before time t and to an edge active after time t, taken over all times t in the lifetime of the graph.

Definition 3 Let (G, τ) be a temporal graph with lifetime Λ. The *vertex interval-membership sequence* of (G, τ) is the sequence $(F_t)_{t \in [\Lambda]}$ of vertex-subsets of G (called *bags*) where each F_t is defined as follows

$$F_t := \{v \in V(G) : \exists i \leq t \leq j \text{ and } u, w \in N_G(v) \text{ such that } i \in \tau(uv) \text{ and } j \in \tau(wv)\}$$

(note that u and w need not be distinct). The *vertex-interval-membership-width* of (G, τ) – denoted $\mathbf{vimw}(G, \tau)$ – is the maximum cardinality attained by any bag in the vertex interval-membership sequence of (G, τ) (i.e. $\mathbf{vimw}(G, \tau) := \max_{t \in [\Lambda]} |F_t|$).

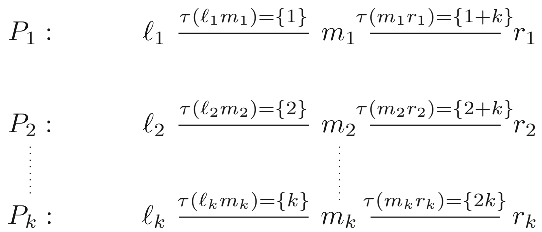

Fig. 4 A temporal graph of unit interval-membership-width, but which has vertex-interval-membership-width at least k; this graph consists of a disjoint union of two-edge paths $P_1, \ldots, P_k$, where the edges $\ell_i m_i$ and $m_i r_i$ of P_i are active at times i and $i+k$ respectively

It is clear that, for any temporal graph (G,τ), $\mathbf{vimw}(G,\tau) \geq 2\mathbf{imw}(G,\tau)$: if an edge e is active at times before and after t, and so belongs to the set for time t in the interval-membership sequence, it follows that both endpoints of e must belong to the set for time t in the vertex interval-membership sequence. However, the difference between the two parameters can be arbitrarily large. Consider, for example, the temporal graph in Fig. 4 consisting of a disjoint union of two-edge paths $P_1, \ldots, P_k$; in this temporal graph, since every edge appears at a single unique time, the interval-membership-width is only one, but the midpoint of every path is incident with an edge appearing at a time at most k and a time strictly greater than k, giving vertex-interval-membership-width at least k.

Lemma 16 *If (G,τ) is a temporal graph where each edge is active at-least once, then one can compute the vertex-interval-membership sequence of (G,τ) in time $\mathcal{O}(\mathbf{vimw}(G,\tau)\Lambda)$.*

Proof Initialize a sequence $(W_t)_{t\in[\Lambda]}$ of empty bags; then proceed as follows:

V1 For each edge e of G, find the pair $(a_e, A_e) = (\min \tau(e), \max \tau(e))$;
V2 for each vertex x of G,

(a) let $(b_x, B_x) = (\min_{e\ni x} a_e, \max_{e\ni x} A_e)$
(b) for all times t with $b_x \leq t \leq B_x$, add x to the bag W_t.

Correctness follows trivially from the definition of the vertex-interval-membership sequence, so now we consider the running time. Denoting the interval-membership sequence of (G,τ) by $(F_t)_{t\in[\Lambda]}$, then, since $2\mathbf{imw}(G,\tau) \leq \mathbf{vimw}(G,\tau)$ (as we observed earlier), Step **V1** takes time

$$\mathcal{O}\Big(\sum_{e\in E(G)} |\tau(e)|\Big) = \mathcal{O}\Big(\sum_{t\in[\Lambda]} |F_t|\Big) = \mathcal{O}(\mathbf{imw}(G,\tau)\Lambda) = \mathcal{O}(\mathbf{vimw}(G,\tau)\Lambda).$$

Step **V2** takes time

$$\begin{aligned}
&\mathcal{O}\Big(\sum_{x\in V(G)} (d_G(x) + |\{t\in[\Lambda] : x\in W_t\}|)\Big) = \\
&= \mathcal{O}(|E(G)|) + \mathcal{O}\Big(\sum_{x\in V(G)} |\{t : x\in W_t\}|\Big) \\
&= \mathcal{O}\Big(\sum_{t\in[\Lambda]} |F_t|\Big) + \mathcal{O}\Big(\sum_{x\in V(G)} |\{t : x\in W_t\}|\Big) \qquad (\forall e\in E(G),\ |\tau(e)|\geq 1)
\end{aligned}$$

$$
\begin{aligned}
&= \mathcal{O}\Big(\sum_{t\in[\Lambda]} |F_t|\Big) + \mathcal{O}\Big(\sum_{t\in[\Lambda]} |W_t|\Big)\\
&= \mathcal{O}(\mathbf{imw}(G,\tau)\Lambda) + \mathcal{O}(\mathbf{vimw}(G,\tau)\Lambda)\\
&= \mathcal{O}(\mathbf{vimw}(G,\tau)\Lambda) && (\mathbf{imw}(G,\tau) \le \mathbf{vimw}(G,\tau)).
\end{aligned}
$$

Thus, the whole algorithm runs in $\mathcal{O}(\mathbf{vimw}(G,\tau)\Lambda)$ time, as desired. □

We now illustrate the greater algorithmic power of vertex-interval-membership-width by showing that MinReachDelete admits an FPT algorithm with respect to this larger parameter.

Theorem 17 *There is an algorithm that decides whether any temporal graph* (G,τ) *with lifetime* Λ *is a yes-instance of* MinReachDelete *in time* $\mathcal{O}(w^2 h 2^{w^2}\Lambda)$*, where* $w = \mathbf{vimw}(G,\tau)$ *is the vertex-interval-membership-width of* (G,τ).

Proof We proceed by a dynamic programming argument similar to that of Theorem 7. Let $((G,\tau), S, k, h)$ be an instance of MinReachDelete, let $(F_t)_{t\in[\Lambda]}$ be the vertex interval-membership sequence of (G,τ) and assume without loss of generality that F_1 is non-empty.

For each $i \in [\Lambda]$, we compute a set $L_i \subseteq \{0,\dots,h\} \times F_i^{\{0,1\}}$ consisting of pairs of the form (r, f) where r is an integer between 0 and h and f is a function from F_i to $\{0,1\}$. We define L_i to be the set of such pairs (r, f) such that there exists a set E_i' of time-edges with the following properties:

- $|E_i'| \le k$, and
- if R denotes the set of vertices reachable from S in $(G,\tau) \setminus E_i$ no later than time i, then
  - $|R| \le r$, and
  - for every $v \in F_i$, $v \in R$ if and only if $f(v) = 1$.

We say that such a set E_i' of time-edges *witnesses* the element (r, f) for L_i. Note that we may assume without loss of generality that every time-edge in E_i' appears at a time less than or equal to i. For each $(r, f) \in L_i$, we set the *cost* of (r, f), written $\mathbf{cost}_i(r, f)$, to be the minimum cardinality of any set of time-edges witnessing (r, f) for L_i. It is clear from these definitions that we have a yes-instance for MinReachDelete if and only if $L_\Lambda \neq \emptyset$, and that in this case the minimum number of edges we need to delete is equal to $\min_{(r,f)\in L_\Lambda} \mathbf{cost}_\Lambda(r, f)$.

It therefore remains to demonstrate that we can compute all sets L_i within the stated time; in fact, we shall also compute the cost function for each L_i as we will make use of this in computing L_{i+1}. Note first that, for all $i \in [\Lambda]$, $|L_i| \le h2^w$. Moreover, recall from the definition of the vertex interval-membership sequence that both endpoints of every edge active at time i belong to F_i, and so the number of edges active at time i is at most $\binom{|F_i|}{2} \le |F_i|^2$.

We begin by considering L_1, and let E_1 be the set of time-edges appearing at time one. Since $|E_1| \le |F_1|^2 \le w^2$, we can consider all 2^{w^2} possibilities for a subset $E_1' \subseteq E_1$ to delete; for each such subset of cardinality at most k, it is clear that we can compute in time $\mathcal{O}(|E_1|) = \mathcal{O}(w^2)$ the corresponding pair (r, f) and record $|E_1'|$ as an upper bound for $\mathbf{cost}_1(r, f)$, since every vertex outside S that is reachable from S by time 1 must belong to F_1. We can therefore compute L_1 in time $\mathcal{O}(w^2 2^{w^2})$.

Now supposing that we have computed L_i and the function $\mathbf{cost}_i : L_i \to \{0, \dots, k\}$, we explain how to compute L_{i+1} and the function $\mathbf{cost}_{i+1} : L_{i+1} \to \{0, \dots, k\}$. Observe that every vertex reached from S by time $i + 1$ that was not already reached by time i must belong to F_{i+1}: such a vertex must be reached along an edge appearing at time $i + 1$ and so is incident with such an edge. Moreover, any such vertex v is reached via an element of $F_i \cup S$: either v is reached directly from S via an edge active at time $i + 1$, or v is reached from another vertex u, which was reached by time i; since u is therefore incident with an edge appearing at a time at most i and an edge active at time $i + 1$, we conclude that $u \in F_i \cap F_{i+1}$.

It follows that the set of vertices reachable from S by time $i + 1$ can be computed from the set of vertices in $F_i \cap F_{i+1}$ reachable from S by time i together with the set of edges active at time $i + 1$. Writing E_{i+1} for the number of edges active at time $i + 1$, we have $|E_{i+1}| \leq \binom{w}{2}$, and so we can consider each of the $2^{\binom{w}{2}}$ possibilities for which edges in this set to delete. For each subset $E_{i+1}' \subseteq E_{i+1}$ and for every state $(r, f) \in L_i$, set

$$\begin{aligned} R_{f,E_{i+1}'} :=& \{v \in F_{i+1} : \exists s \in S \text{ with } sv \in E_{i+1} \setminus E_{i+1}'\} \\ & \cup \{v \in F_{i+1} : \exists u \in F_i \text{ with } f(u) = 1 \text{ and } uv \in E_{i+1} \setminus E_{i+1}'\}. \end{aligned}$$

By the previous reasoning, it is clear that $R_{f,E_{i+1}'}$ is precisely the set of vertices reached at time exactly $i + 1$ under the assumptions that

1. $f^{-1}(1)$ is the set of vertices in F_i reachable by time i, and
2. E_{i+1}' is the set of time-edges deleted at time $i + 1$.

For each subset $E_{i+1}' \subseteq E_{i+1}$ and for every state $(r, f) \in L_i$ such that

1. $\mathbf{cost}_i(r, f) + |E_{i+1}'| \leq k$, and
2. $r + |R_{f,E_{i+1}'}| \leq h$,

we therefore add to L_{i+1} the state (r', f'), where $r' = r + |R_{f,E_{i+1}'}|$ and, for all $v \in F_{i+1}$,

$$f'(v) = \begin{cases} 1 & \text{if } v \in R_{f,E_{i+1}'} \\ 0 & \text{otherwise.} \end{cases}$$

We also record $\mathbf{cost}_i(r, f) + |E_{i+1}'|$ as an upper bound for $\mathbf{cost}_{i+1}(r', f')$. After iterating through all possibilities for E_{i+1}' and (r, f), the true value of $\mathbf{cost}_{i+1}(r', f')$ is the least upper bound we have recorded for this state in L_{i+1}.

It remains only to bound the time needed to compute L_{i+1} and the associated cost function as described. To do this, we consider each pair consisting of a subset E_{i+1}' of time-edges to delete, for which there are $\mathcal{O}(2^{\binom{w}{2}})$ possibilities, together with a state $(r, f) \in L_i$, for which there are $\mathcal{O}(h2^w)$. The total number of pairs we consider is therefore $\mathcal{O}(h2^{w^2})$. For each such pair, we can compute the corresponding state in L_{i+1} and the upper bound on the associated cost by examining each of the edges in $E_{i+1} \setminus E'_{i+1}$, of which there are $\mathcal{O}(w^2)$. We therefore compute L_{i+1} and the function $\mathbf{cost}_{i+1}$ in time $\mathcal{O}(w^2 h 2^{w^2})$.

Summing over all sets L_i for $1 \leq i \leq \Lambda$ we see that the time needed to compute all states and associated cost functions is $\mathcal{O}(w^2 h 2^{w^2} \Lambda)$, as required. □

7 Discussion

In this paper we introduced a natural temporal analogue of Eulerian circuits and proved that, in contrast to the static case, TempEuler(k) is NP-complete for all $k \geq 2$ (where the $k = 2$ case was proven by Marino and Silva [19]). In fact we showed that the problem remains hard even when the underlying static graph has path-width 2, feedback vertex number 1 or vertex cover number 2 (Sect. 3). Along the way, we resolved an open problem of Akrida, Mertzios and Spirakis [15] by showing that StarExp(k) is NP-complete for all $k \geq 4$. This result yields a complete complexity dichotomy with respect to k when combined with Akrida, Mertzios and Spirakis' results [15].

Our hardness results rule out FPT algorithms for TempEuler(k) and StarExp(k) with respect to many standard parameters describing the structure of the underlying graph (such as path-width, feedback vertex number and vertex-cover number). Motivated by these results, we introduced a new width measure which captures structural information that is purely temporal; we call this the *interval-membership-width*. In contrast to our hardness results, we showed that TempEuler(k) and StarExp(k) can be solved in times $\mathcal{O}(w^3 2^w \Lambda)$ and $\mathcal{O}(w^3 2^{3w} \Lambda)$ respectively where w is our new parameter and Λ is the lifetime of the input.

Our fixed-parameter-tractability results parameterized by interval-membership-width can also be leveraged via a win-win approach to obtain tractability results for both TempEuler(k) and StarExp(k) parameterized solely by k and the minimum and maximum differences between any two successive times in a time-set of any edge. These results allow us to partially resolve another open problem of Akrida, Mertzios and Spirakis concerning the complexity of StarExp(k): we showed that it can be solved in time $2^{\mathcal{O}(k)}\Lambda$ when the input has evenly spaced appearances of each edge and lifetime Λ. We note, however, that it remains an open problem to determine the complexity of the evenly spaced StarExp(k) problem when k is unbounded.

Given the success of parameterizations by interval-membership-width when it comes to temporal edge-exploration problems, it is natural to ask whether such parameterizations can also yield FPT algorithms for problems involving vertex exploration or reachablity. It turns out that, for certain problems of this kind, a more powerful parameter is needed to give fixed-parameter-tractability: we show that the vertex-reachability problem MinReachDelete remains intractable even on temporal graphs which have unit interval-membership-width. This motivated us to introduce a "vertex-variant" of our measure called *vertex-interval-membership-width*. This parameter is bounded below by interval-membership-width, but the difference between the two can be arbitrarily large. Parameterizing by this larger parameter puts MinReachDelete in FPT and demonstrates the greater algorithmic power of vertex-interval-membership-width.

Finally we point out that all of our hardness reductions hold also for the case of non-strict temporal walks and, with slightly more work, even our tractability results can be seen to hold for the non-strict case.

Acknowledgements The authors would like to thank Samuel Hand and John Sylvester for spotting slight inaccuracies in the preliminary version of this article and the anonymous reviewers for their helpful comments and suggestions.

Declarations

Funding The first author was supported by an EPSRC doctoral training account and also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 803421, ReduceSearch). The second author was supported by a Royal Society of Edinburgh Personal Research Fellowship, funded by the Scottish Government, and EPSRC grant EP/T004878/1.

References

1. Bumpus, B.M., Meeks, K.: Edge exploration of temporal graphs. In: Flocchini, P., Moura, L. (eds.) Combinatorial Algorithms - 32nd International Workshop, IWOCA 2021. Lecture Notes in Computer Science, vol. 12757, pp. 107–121. Springer (2021)
2. Akrida, E.C., Gąsieniec, L., Mertzios, G.B., Spirakis, P.G.: The complexity of optimal design of temporally connected graphs. Theory Comput. Syst. **61**(3), 907–944 (2017)
3. Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N.: Time-varying graphs and dynamic networks. Int. J. Parallel Emergent Distrib. Syst. **27**(5), 387–408 (2012)
4. Himmel, A.-S., Molter, H., Niedermeier, R., Sorge, M.: Adapting the bron-kerbosch algorithm for enumerating maximal cliques in temporal graphs. Soc. Netw. Anal. Min. **7**(1), 35 (2017)
5. Holme, P., Saramäki, J.: Temporal networks. Phys. Rep. **519**(3), 97–125 (2012)
6. Michail, O.: An introduction to temporal graphs: An algorithmic perspective. Internet Math. **12**(4), 239–280 (2016)
7. Akrida, E.C., Mertzios, G.B., Nikoletseas, S., Raptopoulos, C., Spirakis, P.G., Zamaraev, V.: How fast can we reach a target vertex in stochastic temporal graphs? J. Comput. Syst. Sci. **114**, 65–83 (2020)
8. Axiotis, K., Fotakis, D.: On the Size and the Approximability of Minimum Temporally Connected Subgraphs. In: Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D. (eds.) 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016). Leibniz International Proceedings in Informatics (LIPIcs), vol. 55, pp. 149–114914. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2016)
9. Bhadra, S., Ferreira, A.: Computing multicast trees in dynamic networks and the complexity of connected components in evolving graphs. J. Internet Serv. Appl. **3**(3), 269–275 (2012)
10. Casteigts, A., Himmel, A., Molter, H., Zschoche, P.: Finding temporal paths under waiting time constraints. Algorithmica **83**(9), 2754–2802 (2021)

11. Mertzios, G.B., Michail, O., Spirakis, P.G.: Temporal network optimization subject to connectivity constraints. Algorithmica **81**(4), 1416–1449 (2019)
12. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. J. Comput. Syst. Sci. **64**(4), 820–842 (2002)
13. Wu, H., Cheng, J., Ke, Y., Huang, S., Huang, Y., Wu, H.: Efficient algorithms for temporal path computation. IEEE Trans. Knowl. Data Eng. **28**(11), 2927–2942 (2016)
14. Xuan, B.B., Ferreira, A., Jarry, A.: Computing shortest, fastest, and foremost journeys in dynamic networks. Int. J. Found. Comput. Sci. **14**(02), 267–285 (2003)
15. Akrida, E.C., Mertzios, G.B., Spirakis, P.G., Raptopoulos, C.L.: The temporal explorer who returns to the base. J. Comput. Syst. Sci. **120**, 179–193 (2021)
16. Erlebach, T., Hoffmann, M., Kammer, F.: On temporal graph exploration. J. Comput. Syst. Sci. **119**, 1–18 (2021)
17. Michail, O., Spirakis, P.G.: Traveling salesman problems in temporal graphs. Theor. Comput. Sci. **634**, 1–23 (2016)
18. Euler, L.: Solutio problematis ad geometriam situs pertinentis. Commentarii academiae scientiarum Petropolitanae **5**, 128–140 (1741)
19. Marino, A., Silva, A.: Königsberg sightseeing: Eulerian walks in temporal graphs. In: Flocchini, P., Moura, L. (eds.) Combinatorial Algorithms - 32nd International Workshop, IWOCA 2021. Lecture Notes in Computer Science, vol. 12757, pp. 485–500. Springer (2021)
20. Mertzios, G.B., Molter, H., Niedermeier, R., Zamaraev, V., Zschoche, P.: Computing maximum matchings in temporal graphs. In: Paul, C., Bläser, M. (eds.) 37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020. LIPIcs, vol. 154, pp. 27–12714. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
21. Fluschnik, T., Molter, H., Niedermeier, R., Renken, M., Zschoche, P.: As time goes by: Reflections on treewidth for temporal graphs. In: Fomin, F.V., Kratsch, S., van Leeuwen, E.J. (eds.) Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 12160, pp. 49–77. Springer (2020)
22. Hand, S., Enright, J., Meeks, K.: The temporal firefighter problem. arXiv preprint arXiv:2202.12599 (2022)
23. Enright, J., Meeks, K., Molter, H.: Counting temporal paths. arXiv preprint arXiv:2202.12055 (2022)
24. Diestel, R.: Graph Theory. Springer, Heidelberg (2010)
25. Cygan, M., Fomin, F.V., Kowalik, Ł, Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Cham (2015)
26. Berman, K.A.: Vulnerability of scheduled networks and a generalization of menger's theorem. Networks **28**(3), 125–134 (1996)
27. Garey, M.R., Johnson, D.S.: Computers and Intractability: a Guide to the Theory of NP-completeness. W. H. Freeman, San Francisco (1979)
28. Molter, H., Renken, M., Zschoche, P.: Temporal Reachability Minimization: Delaying vs. Deleting. In: Bonchi, F., Puglisi, S.J. (eds.) 46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 202, pp. 76–17615. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021)
29. Enright, J., Meeks, K., Mertzios, G.B., Zamaraev, V.: Deleting edges to restrict the size of an epidemic in temporal networks. J. Comput. Syst. Sci. **119**, 60–77 (2021)