On Computing Optimal Temporal Branchings

On Computing Optimal Temporal Branchings*

Daniela Bubboloni[1], Costanza Catalano[1], Andrea Marino[2], and Ana Silva[3]

[1] Dipartimento di Matematica e Informatica, Università degli Studi di Firenze, Firenze, Italy {daniela.bubboloni,costanza.catalano}@unifi.it
[2] Dipartimento di Statistica, Informatica, Applicazioni, Università degli Studi di Firenze, Firenze, Italy andrea.marino@unifi.it
[3] Departamento de Matematica, Universidade Federal Do Ceara, Fortaleza, Brazil anasilva@mat.ufc.br

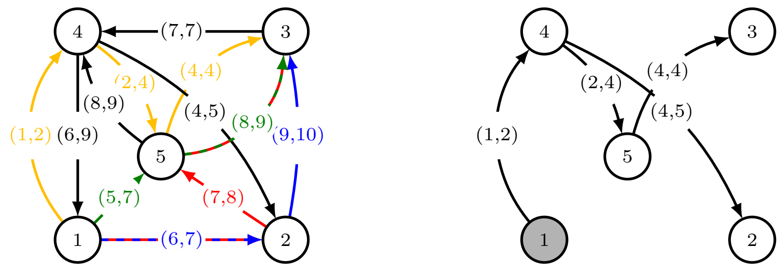
Abstract. The computation of out/in-branchings spanning the vertices of a digraph (also called directed spanning trees) is a central problem in theoretical computer science due to its application in reliable network design. This concept can be extended to temporal graphs, which are graphs where arcs are available only at prescribed times and paths make sense only if the availability of the arcs they traverse is non-decreasing. In this context, the paths of the out-branching from the root to the spanned vertices must be valid temporal paths. While the literature has focused only on minimum weight temporal out-branchings or the ones realizing the earliest arrival times to the vertices, the problem is still open for other optimization criteria. In this work we define four different types of optimal temporal out-branchings (TOB) based on the optimization of the travelling time (ST-TOB), of the travel duration (FT-TOB), of the number of transfers (MT-TOB) or of the departure time (LD-TOB). For $\text{D} \in \{\text{ST},\text{MT},\text{LD}\}$, we provide necessary and sufficient conditions for the existence of spanning D-TOBs; when those do not exist, we characterize the maximum vertex set that a D-TOB can span. Moreover, we provide a log linear algorithm for computing such D-TOBs. Oppositely, we show that deciding the existence of an FT-TOB spanning all the vertices is NP-complete. This is quite surprising, as all the above distances, including FT, can be computed in polynomial time, meaning that computing temporal distances is inherently different from computing D-TOBs. Finally, we show that the same results hold for optimal temporal in-branchings.

Keywords: Temporal graph · temporal network · link stream · optimal branching · optimal temporal walk.

1 Introduction

A temporal graph is a graph where arcs are active only at certain time instants, with a possible *delay* or *travelling time* indicating the time it takes to traverse

* Partially supported by: FUNCAP MLC-0191-00056.01.00/22 and PNE-0112-00061.01.00/16, CNPq 303803/2020-7, Italian PNRR CN4 Centro Nazionale per la Mobilità Sostenibile, NextGeneration EU - CUP B13C22001000001.



(a) $\mathcal{G}$ (b) EA-TOB

Fig. 1: (a) Temporal graph with different walks from vertex 1 to vertex 3, each one represented by a color (two-tone arcs belong to two walks). Yellow: walk realizing EA$(1, 3)$. Green: walk realizing MT$(1, 3)$. Blue: walk realizing both ST$(1, 3)$ and LD$(1, 3)$. Red: walk realizing FT$(1, 3)$. (b) An EA-TOB of $\mathcal{G}$ with root 1.

an arc. There is not a unified terminology in the literature to call these objects, as they are also known as stream graphs [17], dynamic networks [22], temporal networks [15], and time-varying graphs [16] to name a few. Important categories of temporal graphs are those of transport networks, where arcs are labeled by the times of bus/train/flight departures and arrivals [8], and communication networks as phone calls and emails networks, where each arc represents the interaction between two parties [23]. Temporal graphs find application in a vast number of fields such as neural, ecological and social networks, distributed computing, epidemiology etc.; we refer the reader to [12] for a survey on temporal graphs. Fundamental properties of static graphs, as the fact that concatenation of walks is a walk, do not necessarily hold in temporal graphs. For instance, a public transports route can happen only at increasing time instants, since a person cannot catch a bus that already left. This often makes temporal graphs much harder to handle: e.g. computing strongly connected components takes linear time in a static graph, but is an NP-complete problem in a temporal graph [9], and the same happens to Eulerian walks [19], and many other problems. We will see in the next section that this is also the case for temporal branchings.

Background on Temporal Graphs. Given $n \in \mathbb{N}$, we set $[n] := \{x \in \mathbb{N} : x \leq n\}$. A temporal graph $\mathcal{G}$ is a triple (V, A, τ), where V is the set of vertices, $\tau \in \mathbb{N}$ is the *lifetime*, and $A \subseteq \{(u, v, s, t) : u, v \in V, u \neq v \text{ and } s, t \in [\tau], s \leq t\}$ is the set of *temporal arcs*. We set $|A| := M$ and $|V| := n$. Given $a \in A$, we write $a = (\mathrm{t}(a), \mathrm{h}(a), t_s(a), t_a(a))$, where $\mathrm{t}(a)$ and $\mathrm{h}(a)$ are, respectively, the *tail* and *head* vertices of the temporal arc a, and $t_s(a)$ and $t_a(a)$ are, respectively, the *starting time* and the *arrival time* of a. These functions are easily interpreted: $t_s(a)$ is the time at which it is possible to begin a trip along a from vertex $\mathrm{t}(a)$ to vertex $\mathrm{h}(a)$, and $t_a(a)$ is the arrival time of that trip. The temporal graph $\mathcal{G}$ has the multidigraph $\mathcal{D}_{\mathcal{G}} = (V, A, \mathrm{t}, \mathrm{h})$ as underlying structure. Figure (1a) presents an example of temporal graph, where every arc a is labeled by the ordered pair $(t_s(a), t_a(a))$. Each arc has an *elapsed* time $el(a) := t_a(a) - t_s(a)$. In temporal graphs, walks make sense only if they are time-consistent, meaning that each arc

Table 1: Computational time of single source shortest paths in a temporal graph.

EA	MT	ST	LD	FT
$O(M)$ [13, 24]	$O(M \log n)$ [2]	$O(M \log M)$ [1, 24, 25]	$O(M \log M)$ [1]	$O(M \log n)$ [2]

of the walk must have an arrival time smaller or equal than the starting time of the subsequent arc in the walk. More precisely, a *temporal* (u, v)-walk of *length* $k \in \mathbb{N}$ in $\mathcal{G}$ is a (u, v)-walk $W = (u, a_1, v_1, \ldots, v_{k-1}, a_k, v)$ in the underlying multidigraph such that $t_a(a_i) \leq t_s(a_{i+1})$ for all $i \in [k-1]$; in this case we also say that v is *temporally reachable* from u. For the walk W, we consider the *starting time* $t_s(W) := t_s(a_1)$ and the *arrival time* $t_a(W) := t_a(a_k)$. The *travelling time* of W is $\mathrm{tt}(W) := \sum_{i=1}^{k} el(a_i)$ and the *duration* of W is $\mathrm{dur}(W) := t_a(W) - t_s(W)$. The *length* of W is denoted by $\ell(W)$. Given $u, v \in V$, $\mathcal{W}_{\mathcal{G}}(u, v)$ is the set of temporal walks from u to v in $\mathcal{G}$. We consider the following optimization criteria.

Earliest Arrival time: $\textsc{ea}_{\mathcal{G}}(u, v) := \min\{t_a(W) : W \in \mathcal{W}_{\mathcal{G}}(u, v)\}$;
Latest Departure time: $\textsc{ld}_{\mathcal{G}}(u, v) := \max\{t_s(W) : W \in \mathcal{W}_{\mathcal{G}}(u, v)\}$;
Minimum Transfers: $\textsc{mt}_{\mathcal{G}}(u, v) := \min\{\ell(W) : W \in \mathcal{W}_{\mathcal{G}}(u, v)\}$;
Fastest Time: $\textsc{ft}_{\mathcal{G}}(u, v) := \min\{\mathrm{dur}(W) : W \in \mathcal{W}_{\mathcal{G}}(u, v)\}$;
Shortest Travelling time: $\textsc{st}_{\mathcal{G}}(u, v) := \min\{\mathrm{tt}(W) : W \in \mathcal{W}_{\mathcal{G}}(u, v)\}$.

Consistently with the literature [3], we refer to the above definitions as *distances.*[4] All these concepts are widely used (see [1, 2, 8, 13, 24, 25]), although sometimes they appear with different names. For any $\textsc{d} \in \{\textsc{ea}, \textsc{ld}, \textsc{mt}, \textsc{ft}, \textsc{st}\}$, we say that a temporal (u, v)-walk *realizes* $\textsc{d}_{\mathcal{G}}(u, v)$ if it attains the minimum (or maximum if $\textsc{d}=\textsc{ld}$) of the functions in the corresponding definition of $\textsc{d}_{\mathcal{G}}(u, v)$. Figure (1a) shows, for each $\textsc{d}$, a temporal walk from vertex 1 to 3 realizing $\textsc{d}_{\mathcal{G}}(u, v)$. Each distance is computable in polynomial-time: Table 1 reports the time to compute $\textsc{d}_{\mathcal{G}}(r, v)$ from a given vertex r to all the other vertices v.

Optimal temporal branchings. In static directed graphs, spanning branchings are well-studied objects; they represent a minimal set of arcs that connect a special vertex called the root to any other vertex (out-branching), or any vertex to the root (in-branching). They are also called arborescences or spanning directed trees, since their underlying structure is a tree. Spanning branchings representing shortest distances are also well-studied. Their existence is guaranteed simply by the reachability of any vertex from/to the root and they can be computed in $O(M \log M)$ time by Dijkstra's algorithm [7]. Branchings are, to cite a few, important for engineering applications as they represent the cheapest or shortest way to reach all vertices, and in social networks in relation to information dissemination and spreading. We can similarly define spanning branchings in temporal graphs, here called spanning TOBs (Temporal Out-Branchings) and TIBs (Temporal In-Branchings), representing the minimal set of temporal arcs that temporally connect any vertex from/to the root. This definition of TOB has

[4] Notice that they do not necessarily satisfy the triangle inequality.

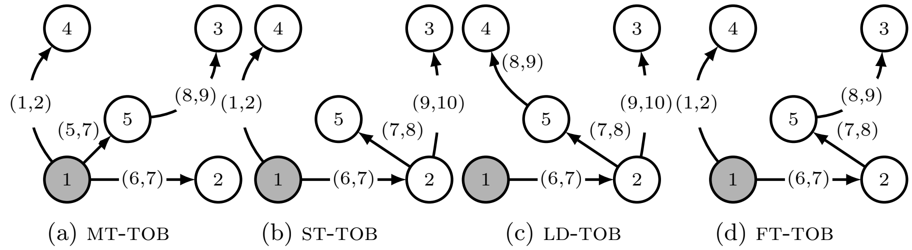
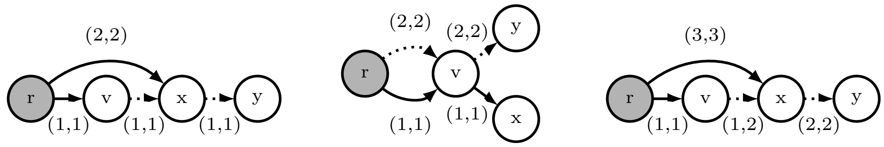


Fig. 2: Example of D-TOBs of the temporal graph $\mathcal{G}$ in Figure (1a) for different distances. The grey vertex is the root of the TOB.

already appeared in the literature [13, 14].[5] In the context of urban mobility, suppose that a festival or a concert is just finished in a remote location X late at night, and you want to guarantee that every person can go back home via public transports, while optimizing the number of bus/train rides. This problem can be solved by a spanning TOB with root X. We also may ask this TOB to arrive the earliest possible in every point of interest of the city, or to use the least number of transfers, or optimize any of the distances that we have introduced before. It is then natural to extend the notion of shortest distance branchings to the temporal framework. For each $\text{D} \in \{\text{EA}, \text{LD}, \text{MT}, \text{FT}, \text{ST}\}$, we call spanning D-TOB a spanning TOB representing the distance D, i.e. for every vertex v, the unique (r, v)-walk within the branching realizes $\text{D}(r, v)$. We define similarly spanning D-TIBs. Figure (1b) and Figure 2 show, for each $\text{D} \in \{\text{EA}, \text{MT}, \text{ST}, \text{LD}, \text{FT}\}$, a spanning D-TOB with root 1 of the temporal graph in Figure (1a). Notice that the MT-TOB can be modified by adding the arc $(9, 10)$ from vertex 2 to vertex 3 and by deleting the arc $(8, 9)$ from vertex 5 to vertex 3 and still obtaining a spanning MT-TOB. Thus, in general, D-TOBs are not unique. In [13] the authors prove that a spanning TOB as well as a EA-TOB exist iff every vertex is temporally reachable from the root and provide an algorithm to compute them in $O(M)$ time. Nonetheless, for all the other distances but EA, the problem of computing optimal branching is still open and seems to be a more difficult task. We start observing that for $\text{D} \neq \text{EA}$, the temporal reachability from the root to any vertex is no longer sufficient for the existence of a spanning D-TOB; this is showed in Figure 3 where for each $\text{D} \in \{\text{LD}, \text{MT}, \text{ST}, \text{FT}\}$ we present a temporal graph that does not admit a spanning D-TOB even if every vertex is temporally reachable from r. In Figures (3a) and (3c), observe that there is a unique temporal path from r to y; call it P. This is clearly the only spanning TOB of the temporal graphs under consideration. However, P does not realize $\text{D}(r, x)$, which is realized by the temporal arc from r to x. Therefore, P is not a D-TOB. We emphasize that adding the arc from r to x to P would no longer form a TOB (the underlying graph would not be a branching). As for Figure (3b), notice that there is a unique temporal path that realizes $\text{FT}(r, x)$, namely the one made of

[5] Notice that [14] proposes it in a simplified context, while the conditions listed in the definition of [13] are not all necessary to describe the concept (see Lemma 1).



(a) No spanning MT,LD-TOB (b) No spanning FT-TOB (c) No spanning ST-TOB

Fig. 3: Examples of temporal graphs that do not admit a spanning D-TOB with root r. Solid arcs represent a maximum D-TOB.

the temporal arcs $(r, v, 1, 1)$ and $(v, x, 1, 1)$. Similarly, there is a unique temporal path that realizes $\text{FT}(r, y)$, namely the one made of the temporal arcs $(r, v, 2, 2)$ and $(v, y, 2, 2)$. This implies that a possible spanning D-TOB must be equal to the graph itself, which clearly is not a branching. Notice that in the examples, $\tau = 2$ for $\text{D} \in \{\text{MT}, \text{LD}\}$, which is the smallest value possible, as when $\tau = 1$ the temporal graph reduces to a static graph. When $\text{D} = \text{ST}$, we have that $\tau = 3$: it can be proven that this is again the smallest value possible (see Appendix F). Notice also that in all the examples, we can always find a D-TOB on the vertex set equal to $\{r, v, x\}$, with D chosen accordingly; this TOB is highlighted by solid arcs in the figures. In Figure (3b), also the dotted arcs form an FT-TOB on the vertex set $\{r, v, y\}$. The following questions naturally arise:

1. When does a spanning D-TOB exist?
2. If it does not exist, can we identify the maximum set of vertices that can be spanned by a D-TOB (*maximum* D-TOB)?
3. Can we compute a maximum D-TOB in polynomial time?
4. Can we answer to all the above questions for D-TIBs?

In this paper we solve all these problems.

Our contribution. We first show some characterizations of TOBs. Each of them gives a different insight on these objects. Then, for each $\text{D} \in \{\text{ST}, \text{MT}, \text{LD}\}$, we provide a necessary and sufficient condition for the existence of a spanning D-TOB in a temporal graph; this property is based on the concept of optimal substructure. Moreover, we characterize the vertex set of a maximum D-TOB, which turns out to be uniquely identified; this property is crucial to find efficient polynomial-time algorithms for computing a maximum D-TOB (Section 4). In particular, our algorithms compute a D-TOB whose path from the root arrives the earliest possible in every vertex. The characterization does not hold for $\text{D} = \text{FT}$, and in fact we show that computing an FT-TOB is an NP-complete problem (Section 5). Finally, in Section 3, we show that the same results hold for optimal temporal in-branchings. A summary of our results and of the computational time of our algorithms can be found in Table 2. We underline that any algorithm computing $\text{D}(r, v)$ for all vertices v of a temporal graph cannot suffice by itself to find a D-TOB. Indeed we have seen in Figure (3a) and (3c) that $\text{D}(r, y)$ is well-defined because y is temporally reachable from the root r, but no D-TOBs can span y. In other words, there are no guarantees that the union of the shortest paths,

Table 2: Our contribution: summary results. The second column refers to the time to compute any TOB or TIB, while the others refer to the time to compute any D-TOB or D-TIB for D $\in$ {EA,MT,ST,LD,FT}.

	any	EA-	MT-	ST-	LD-	FT-
TOB	$O(M)$ [13]	$O(M)$ [13]	$O(M \log n)$	$O(M \log M)$	$O(M \log M)$	NP-c
TIB	$O(M)$	$O(M \log M)$	$O(M \log n)$	$O(M \log M)$	$O(M)$	NP-c

with respect to the considered distance D, computed by the aforementioned algorithms would form a TOB. In addition, for D =FT we have the extreme case where computing FT(r, v) is polynomial-time, but finding an FT-TOB is NP-complete. Also, applying Dijkstra's algorithm on the static expansion of a temporal graph returns a branching on the static expansion, but does not guarantee to obtain a TOB in the original temporal graph (see Appendix A).

Further Related Results. We have already mentioned the results of [13], where they also show that the problem of finding minimum weight spanning TOBs is NP-hard. Kuwata et. al. [16] are interested in the temporal reachability from the root that realizes the earliest arrival time, and they obtain it by making use of Dijsktra's algorithm on the static expansion of the temporal graph, which we already observed does not translate into a TOB in the original temporal graph (Appendix A). Gunturi et. al. [11] present a polynomial-time algorithm for computing minimum weight TOBs: in their model, the weight of the arcs depend on a function that evolves in time, but walks are not required to be time-respecting. Different versions of the problem of finding arc-disjoint TOBs in temporal graphs are investigated in [4, 14].

2 Preliminaries

We denote by $\mathbb{N}$ the set of positive integers. We set $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, $[n] := \{x \in \mathbb{N} : x \leq n\}$ and $[n]_0 := \{x \in \mathbb{N}_0 : x \leq n\}$, for $n \in \mathbb{N}_0$. Given a set $\mathcal{X}$ and a property $\mathcal{P}$, we say that $\mathcal{X}$ is *minimal* for property $\mathcal{P}$ if $\mathcal{X}$ has property $\mathcal{P}$, and for all $\mathcal{Y} \subsetneq \mathcal{X}$, $\mathcal{Y}$ does not have property $\mathcal{P}$. We remind that a digraph is a pair $D = (V, A)$ where V is the nonempty and finite set of vertices, and $A \subseteq V \times V$ is the set of arcs. Informally, a multidigraph is a digraph where multiple arcs are allowed. A multidigraph is formalized by a quadruple $\mathcal{D} = (V, A, \mathrm{t}, \mathrm{h})$, where V is the set of vertices, A the set of arcs and $\mathrm{t}, \mathrm{h} : A \to V$ are respectively the *head* and the *tail* function, where we require that $\forall a \in A$, $\mathrm{t}(a) \neq \mathrm{h}(a)$, i.e. no selfloops are allowed[6]. The in-neighborhood and out-neighborhood of a vertex v are defined as $N^-_{\mathcal{D}}(v) := \{u : \exists a \in A \text{ s.t. } \mathrm{t}(a) = u, \mathrm{h}(a) = v\}$ and $N^+_{\mathcal{D}}(v) := \{u : \exists a \in A \text{ s.t. } \mathrm{t}(a) = v, \mathrm{h}(a) = u\}$. The in-degree and out-degree of v are defined respectively as $d^-_{\mathcal{D}}(v) := |\{a \in A : \mathrm{h}(a) = v\}|$, $d^+_{\mathcal{D}}(v) := |\{a \in A : \mathrm{t}(a) = v\}|$. A (u, v)-*walk* of length $k \in \mathbb{N}_0$ in $\mathcal{D}$ is an alternating ordered sequence $W = (u, a_1, v_1, \ldots, v_{k-1}, a_k, v_k = v)$ of vertices $u, v_1, \ldots, v_k \in V$ and

[6] Notice that if t and h are injective, $\mathcal{D}$ is a digraph.

arcs $a_1, \dots, a_k \in A$ such that $\mathrm{t}(a_1) = u$, $\mathrm{h}(a_k) = v$ and $\mathrm{h}(a_i) = v_i = \mathrm{t}(a_{i+1})$ for all $i \in [k-1]$. The set of vertices of W is denoted by $V(W)$ and the set of arcs of W by $A(W)$. A *path* is a walk where the vertices are all distinct. A walk W *traverses* an arc a if $a \in A(W)$. For $h \in [k]$ the v_h-prefix of W is the subwalk of W given by $(u, a_1, v_1, \dots, v_h)$; the v_h-suffix of W is the subwalk of W given by $(v_h, a_{h+1}, \dots, a_k, v_k)$. Note that, for a fixed $z \in V(W)$, there are, in general, many z-prefixes and many z-suffixes of W; they are unique if W is a path. Given a (u, v)-walk W and a (v, s)-walk Z, we denote the walk obtained by their concatenation by $W + Z$. For $V' \subseteq V$, the *multidigraph induced* by V' in the multidigraph $\mathcal{D}$ is denoted by $\mathcal{D}[V']$. A digraph $\mathcal{D} = (V, A)$ is called an *out-branching* (resp. *in-branching*) with root $r \in V$ if for every $v \in V$ there exists a unique (r, v)-*walk* (resp. (v, r)-*walk*) in $\mathcal{D}$. Note that in a branching, every walk is a path. When using concepts like in-neighborhood, out-neighborhood, in-degree and out-degree for a temporal graph $\mathcal{G}$, it is intended that we are referring to its underlying multidigraph $\mathcal{D}_{\mathcal{G}}$. From every temporal walk it is possible to extract a temporal path with the same extremes. A temporal graph $\mathcal{G}' = (V', A', \tau')$ is a *temporal subgraph* of $\mathcal{G} = (V, A, \tau)$ if $V' \subseteq V$, $A' \subseteq A$ and $\tau' \leq \tau$. When the temporal graph is clear from the context, we usually omit the subscripts.

3 Temporal Branching and Preliminary Results

3.1 Temporal Out-Branching

In this section, we present the formal notion of temporal out-branching, give some useful characterizations, and define related optmization problems.

Definition 1. *A temporal graph $\mathcal{T} = (V, A, \tau)$ is called a* temporal out-branching (TOB) *with root $r \in V$ if A is a minimal set of temporal arcs such that for all $v \in V$, there exists a temporal (r, v)-walk in $\mathcal{T}$.*

The following lemma provides characterizations of a TOB, which are crucial for the proofs of the results of Section 4.

Lemma 1. *Let $\mathcal{T} = (V, A, \tau)$ be a temporal graph. The following facts are equivalent:*[7]

1. *$\mathcal{T}$ is a TOB with root r;*
2. *For all $v \in V$ there is a temporal (r, v)-walk in $\mathcal{T}$. Additionally, $d^-_{\mathcal{T}}(r) = 0$ and, for all $v \in V \setminus \{r\}$, $d^-_{\mathcal{T}}(v) = 1$;*
3. *For all $v \in V$ there is a temporal (r, v)-walk in $\mathcal{T}$, and $|A| = |V| - 1$;*
4. *The underlying digraph $\mathcal{D}_{\mathcal{T}}$ of $\mathcal{T}$ is an out-branching with root r and for all $v \in V$, the unique (r, v)-walk in $\mathcal{T}$ is temporal.*

Proof. See Appendix B.

In a TOB with root r, the unique temporal walk from r to v is a temporal path.

[7] In [13], Properties 2 and 3 are not recognized as equivalent.

Definition 2. *A temporal graph $\mathcal{G} = (V, A, \tau)$* admits *a* TOB *with root r if there exists a temporal subgraph $\mathcal{T} = (V_\mathcal{T}, A_\mathcal{T}, \tau_\mathcal{T})$ of $\mathcal{G}$ that is a* TOB *with root r. Such a $\mathcal{T}$ is said a* TOB *of $\mathcal{G}$. $\mathcal{T}$ is called a* spanning TOB *of $\mathcal{G}$ if $V_\mathcal{T} = V$; a* maximum TOB *of $\mathcal{G}$ if $|V_\mathcal{T}|$ is the largest possible.*

We now expand the concept of TOB to the various distances considered in the introduction. The idea is that we are not only interested in temporally reaching the maximum number of vertices from the root, but we want also to *minimize* their distance from the root, which can translate into arriving the earliest possible, the fastest possible, by starting the journey the latest possible, by travelling the shortest time possible or by making the least number of transfers possible, depending on the preferences and needs.

Definition 3. *Let* D $\in \{$EA, LD, MT, FT, ST$\}$ *and let $\mathcal{T} = (V_\mathcal{T}, A_\mathcal{T}, \tau_\mathcal{T})$ be a* TOB *with root r of a temporal graph $\mathcal{G} = (V, A, \tau)$. We say that $\mathcal{T}$ is a* D-TOB *of $\mathcal{G}$ if* $\mathrm{D}_\mathcal{T}(r, v) = \mathrm{D}_\mathcal{G}(r, v)$ *for every $v \in V_\mathcal{T}$. $\mathcal{T}$ is a* spanning D-TOB *of $\mathcal{G}$ if $V_\mathcal{T} = V$; is a* maximum D-TOB *of $\mathcal{G}$ if $|V_\mathcal{T}|$ is the largest possible.*

Observe that Lemma 1 and Definition 3 imply the following remark.

Remark 1. A temporal graph $\mathcal{G}$ admits a TOB $\mathcal{T} = (V_\mathcal{T}, A_\mathcal{T}, \tau_\mathcal{T})$ with root r iff every $v \in V_\mathcal{T}$ is temporally reachable from r in $\mathcal{G}$. Moreover, if $\mathcal{T}$ is a D-TOB of $\mathcal{G}$ then $\mathcal{T}$ is a spanning D-TOB of $\mathcal{G}[V_\mathcal{T}]$.

Problem 1 (Maximum D-TOB*).* Let D $\in \{$EA, LD, MT, ST, FT$\}$ and $\mathcal{G}$ be a temporal graph. Find a maximum D-TOB of $\mathcal{G}$.

Problem 1 has already been solved for D = EA in [13]. Their result also implies that a maximum EA-TOB of $\mathcal{G}$ spans all the vertices that are temporally reachable from the root. We will see that also for every D $\in \{$LD, MT, ST$\}$, the vertex set of a maximum D-TOB of a temporal graph is uniquely determined, which is key for the polynomiality of the related problems (Section 4). However, the property of being temporally reachable from the root is not sufficient anymore, as showed in Figure 3. Instead for FT, we show that the related problem is NP-complete (Section 5). As we will see, in the polynomial cases we can constrain ourselves to the earliest arrival paths that realize the distances. For this, we define:

Definition 4. *Given a temporal graph $\mathcal{G} = (V, A, \tau)$, for any $u, v \in V$ and* D $\in$ $\{$MT, ST, LD, FT$\}$, *we define* $\mathrm{EAD}_\mathcal{G}(u, v) := \min\{t_a(W) : W \text{ realizes } \mathrm{D}_\mathcal{G}(u, v)\}$*. A* TOB *$\mathcal{T} = (V_\mathcal{T}, A_\mathcal{T}, \tau_\mathcal{T})$ with root r of $\mathcal{G}$ is an* EAD-TOB *if it is a* D-TOB *and, for every $v \in V_\mathcal{T}$, we have that* $\mathrm{EAD}_\mathcal{T}(r, v) = \mathrm{EAD}_\mathcal{G}(r, v)$*. $\mathcal{T}$ is called* spanning *if $V_\mathcal{T} = V$;* maximum *if $|V_\mathcal{T}|$ is the largest possible.*

3.2 Temporal In-Branching

In this section, we present definitions of temporal in-branchings and prove that the related problems are computationally equivalent to TOBs.

Definition 5. *A temporal graph $\mathcal{T} = (V_\mathcal{T}, A_\mathcal{T}, \tau_\mathcal{T})$ is called a* temporal in-branching (TIB) *with root r if $A_\mathcal{T}$ is a minimal set of temporal arcs such that for all $v \in V$, there exists a temporal (v, r)-walk in $\mathcal{T}$. A temporal graph $\mathcal{G} = (V, A, \tau)$* admits *a* TIB *with root r if there exists a temporal subgraph $\mathcal{T}$ of $\mathcal{G}$ that is a* TIB *with root r; we also say that $\mathcal{T}$ is a* TIB *of $\mathcal{G}$. $\mathcal{T}$ is* spanning *if $V_\mathcal{T} = V$, and it is* maximum *if $|V_\mathcal{T}|$ is the largest possible. Given* D $\in \{$EA, LD, MT, FT, ST$\}$ *and $\mathcal{T}$ a* TIB *with root r of $\mathcal{G}$, we say that $\mathcal{T}$ is a* D-TIB *of $\mathcal{G}$ if for every $v \in V_\mathcal{T}$, $\mathrm{D}_\mathcal{T}(v, r) = \mathrm{D}_\mathcal{G}(v, r)$. If in addition $V_\mathcal{T} = V$, then $\mathcal{T}$ is a* spanning D-TIB, *and if $|V_\mathcal{T}|$ is the largest possible, then $\mathcal{T}$ is a* maximum D-TIB.

Problem 2 (Maximum D-TIB*).* Let D $\in \{$EA, LD, MT, ST, FT$\}$ and $\mathcal{G}$ be a temporal graph. Find a maximum D-TIB of $\mathcal{G}$.

The next proposition shows that finding maximum TIBs can be reduced to finding maximum TOBs in an auxiliary temporal graph. We define the *reversal* of a temporal graph $\mathcal{G} = (V, A, \tau)$ as the temporal graph $\mathcal{G}^\circlearrowleft = (V, A^\circlearrowleft, \tau)$ where the order of the timesteps is reversed as well as the direction of the arcs. Formally, $A^\circlearrowleft = \{(\mathrm{h}(a), \mathrm{t}(a), \tau - t_a(a) + 1, \tau - t_s(a) + 1) : a \in A\} := \{a^\circlearrowleft : a \in A\}$. A similar transformation has been used e.g. in [3].

Proposition 1. *Given a temporal graph $\mathcal{G}$, it holds that:*

1. *$\mathcal{T}$ is a maximum* EA-TIB *of $\mathcal{G}$ iff $\mathcal{T}^\circlearrowleft$ is a maximum* LD-TOB *of $\mathcal{G}^\circlearrowleft$;*
2. *$\mathcal{T}$ is a maximum* LD-TIB *of $\mathcal{G}$ iff $\mathcal{T}^\circlearrowleft$ is a maximum* EA-TOB *of $\mathcal{G}^\circlearrowleft$;*
3. *For each* D $\in \{$MT, ST, FT$\}$*, $\mathcal{T}$ is a maximum* D-TIB *of $\mathcal{G}$ iff $\mathcal{T}^\circlearrowleft$ is a maximum* D-TOB *of $\mathcal{G}^\circlearrowleft$.*

Proof. See Appendix C.

4 Computing maximum D-TOBs for D $\in$ {MT,ST,LD}

The following concept allows us to establish a necessary and sufficient condition for the existence of a spanning D-TOB with root r in a temporal graph.

Definition 6. *Let $\mathcal{G}$ be a temporal graph and W be a temporal (u, v)-walk in $\mathcal{G}$. For every* D $\in \{$LD, MT, FT, ST$\}$ *we say that:*

- *W is* D-prefix-optimal *if $\forall x \in V(W)$, any x-prefix of W realizes $\mathrm{D}_\mathcal{G}(u, x)$;*
- *W is* EAD-prefix-optimal *if it is* D-prefix-optimal *and $\forall x \in V(W)$, any x-prefix of W realizes $\mathrm{EAD}_\mathcal{G}(u, x)$.*

Theorem 1. *Let $\mathcal{G} = (V, A, \tau)$ be a temporal graph, $r \in V$ and* D $\in \{$LD, MT, ST$\}$*. Then $\mathcal{G}$ admits a spanning* D-TOB *with root r if and only if there exists a* D*-prefix-optimal temporal (r, v)-path in $\mathcal{G}$ for all $v \in V$.*

Proof. See Appendix D.

Notice that Theorem 1 does not hold for $\textsc{d} = \textsc{ft}$. Indeed the temporal graph in Figure (3b) has an $\textsc{ft}$-prefix-optimal path from r to any other vertex, but does not admit a spanning $\textsc{ft-tob}$ as previously observed. We are now ready to characterize the vertex set of a maximum $\textsc{d-tob}$.

Corollary 1. *Let $\mathcal{G} = (V, A, \tau)$ be a temporal graph, $r \in V$, and $\textsc{d} \in \{\textsc{ld}, \textsc{mt}, \textsc{st}\}$. Then a maximum $\textsc{d-tob}$ $\mathcal{T}$ with root r of $\mathcal{G}$ has vertex set:*

$$V_{\mathcal{T}} = \{v \in V : \textit{there exists a } \textsc{d}\textit{-prefix-optimal } (r, v)\textit{-path in } \mathcal{G}\ \}. \tag{1}$$

Proof. See Appendix E.

The next sections present algorithms for finding $\textsc{d-tob}$s of a given temporal graph in polynomial time, when $\textsc{d} \in \{\textsc{mt}, \textsc{st}, \textsc{ld}\}$. In particular, we show that these algorithms always return an $\textsc{ead-tob}$. This implies that for $\textsc{d} \in \{\textsc{mt}, \textsc{st}, \textsc{ld}\}$, the existence of a $\textsc{d}$-prefix-optimal (r, v)-path in $\mathcal{G}$ is equivalent to the existence of an $\textsc{ead}$-prefix-optimal (r, v)-path in $\mathcal{G}$. For $\textsc{d} = \textsc{ft}$ this is no longer true: indeed consider Figure (3b). The only $\textsc{ft}$-prefix-optimal (r, y)-path is $W = (r, (r, v, 2), v, (v, y, 2), y)$, but it is not $\textsc{eaft}$-prefix-optimal: in fact, $\textsc{eaft}(r, v) = 1$ since the path $(r, (r, v, 1), v)$ realizes $\textsc{ft}(r, v)$ and arrives in v at time 1, while W arrives in v at time 2. This difference will be crucial for showing that computing an $\textsc{ft-tob}$ is an NP-complete problem (Section 5). Corollary 1 shows that, even if v is temporally reachable from r, if none of the walks that realize $\textsc{d}(r, v)$ is $\textsc{d}$-prefix-optimal, then no $\textsc{d-tob}$ can span v.

4.1 Algorithm for $\textsc{mt}$

Algorithm 1 computes a maximum $\textsc{mt-tob}$ with root r of a given temporal graph. First observe that, given an $\textsc{mt}$-prefix-optimal temporal (r, v)-walk $W = (r = v_0, a_1, v_1, \ldots, a_k, v_k = v)$, we have that $\textsc{mt}(r, v_i) = \textsc{mt}(r, v_{i+1}) - 1 < \textsc{mt}(r, v_{i+1})$ for all $i \in [k-1]$, i.e. the sequence of distances in any $\textsc{mt}$-prefix-optimal walk is strictly increasing. The main idea of the algorithm is then to compute a priori the $\textsc{mt}$-distances of all vertices from the root, and then build the $\textsc{mt-tob}$ guided by these computed distances, using their strict monotonicity property. More specifically, given $h = \max\{\textsc{mt}(r, v) : v \in V\}$, the algorithm grows an $\textsc{mt-tob}$ starting from the root and adding, at step $i \in [h]$, all the vertices at distance i. During this process, when adding some vertex v, we choose, among its neighbors at distance $i-1$, which one can be the parent of v. To choose the right parent, we look at the incoming temporal arcs having tail in vertices at distance $i - 1$ and we consider only the arcs $a' = (u', v, s', t')$ such that, if $W_{u'}$ is the unique temporal (r, u')-path in the $\textsc{mt-tob}$ built so far, then $s' \geq t_a(W_{u'})$, i.e. the new arc can be concatenated with $W_{u'}$ to obtain a temporal (r, v)-path. Among the arcs fulfilling these constraints, we choose a' minimizing t', the arrival time in v; such arc a' exists if and only if there exists an $\textsc{mt}$-prefix-optimal (r, v)-path in $\mathcal{G}$. We prove that such choice of a' ensures that we are actually representing in the $\textsc{tob}$ a temporal (r, v)-path that realizes the distance $\textsc{mt}(r, v)$ and has the earliest arrival time among the walks realizing such distance, i.e.

Algorithm 1: Computing a maximum MT-TOB of a temporal graph.

Input: A temporal graph $\mathcal{G} = (V, A, \tau)$, and a vertex $r \in V$.
Output: A maximum MT-TOB $\mathcal{T} = (V_{\mathcal{T}}, A_{\mathcal{T}}, \tau_{\mathcal{T}})$ of $\mathcal{G}$ with root r.

$\mathcal{EA}(r) \leftarrow 0; \forall v \in V \setminus \{r\}, \mathcal{EA}(v) \leftarrow +\infty;$
$d(r) \leftarrow 0;\ \forall v \in V \setminus \{r\}, d(v) \leftarrow \mathrm{MT}_{\mathcal{G}}(r, v);$
$V_{\mathcal{T}} \leftarrow \{r\};\ A_{\mathcal{T}} \leftarrow \emptyset;\ \tau_{\mathcal{T}} \leftarrow 0;\ h \leftarrow \max\{d(v) : v \in V\};$
for $i = 1, \ldots, h$ **do**
    for *each* $v \in V$ *such that* $d(v) = i$ **do**
        $S \leftarrow \{(u', v, s', t') \in A : s' \geq \mathcal{EA}(u'),\ d(u') = i - 1\};$
        if $S \neq \emptyset$ **then**
            $a \leftarrow$ choose $(u, v, s, t) \in \arg\min_{(u', v, s', t') \in S} t';$
            $\mathcal{EA}(v) \leftarrow t,\ V_{\mathcal{T}} \leftarrow V_{\mathcal{T}} \cup \{v\},\ A_{\mathcal{T}} \leftarrow A_{\mathcal{T}} \cup \{a\},\ \tau_{\mathcal{T}} \leftarrow \max\{\tau_{\mathcal{T}}, t\};$
        end
    end
end

we are computing an EAMT-TOB. The algorithm takes $O(M \log n)$ time to compute all the initial MT distances (see Table 1), while the remaining part of the algorithm takes $O(M)$ time as it requires only one scan of each temporal arc.

Theorem 2. *Algorithm 1 returns a maximum* MT-TOB *of a temporal graph, for a chosen root, in $O(M \log n)$ time. Besides, the output is an* EAMT-TOB.

Proof. See Appendix G.

4.2 Algorithm for LD and ST

Algorithm 2 computes a maximum D-TOB $\mathcal{T}$ with root r for a given temporal graph when D $\in \{$LD, ST$\}$, and it is more involved with respect to Algorithm 1. The issue is that if $W = (r = v_0, a_1, \ldots, a_k, v_k = v)$ is a D-prefix-optimal walk, then it is possible to have $\mathrm{D}(r, v_{i-1}) = \mathrm{D}(r, v_i)$ for some $i \in [k]$. Indeed, if D $=$ LD, then all the vertices in the walk share the same latest departure time, i.e. $t_s(W) = \mathrm{LD}(r, v_i)$ for all $i \in [k]$. If D $=$ ST and $el(a_i) = 0$ for some $i \in [k]$, then $\mathrm{ST}(r, v_{i-1}) = \mathrm{ST}(r, v_i)$. However, in any case we have that $\mathrm{D}(r, v_{i-1}) \leq \mathrm{D}(r, v_i)$ for all $i \in [k]$. This implies that, by letting D_i denote the set of vertices at distance d_i from r with the distances d_i being in increasing order, to choose the parent of each vertex of D_i in $\mathcal{T}$, we cannot look only at vertices in $D_0 \cup \cdots \cup D_{i-1}$, but also at the ones in D_i itself (in particular, only at the ones in D_i when D $=$ LD). Note that this gives us an additional difficulty as we cannot simply choose an arbitrary vertex $v \in D_i$ to be the next one to be added to $\mathcal{T}$, as it might happen that the good parent of v (i.e. the in-neighbor of v within an EAD-prefix-optimal (r, v)-walk) has not been added to $\mathcal{T}$ yet. To overcome this, we add vertices in D_i to $\mathcal{T}$ in increasing order of the value of $\mathrm{EAD}(r, v)$. Observe however that $\mathrm{EAD}(r, v)$ is not known a priori, so to do that we use a queue that keeps the outgoing temporal arcs from vertices in $\mathcal{T}$ in increasing order of their arrival

Algorithm 2: Computing a maximum D-TOB, with $\text{D} \in \{\text{LD}, \text{ST}\}$.

Input: A temporal graph $\mathcal{G} = (V, A, \tau)$, a vertex $r \in V$, $\text{D} \in \{\text{LD}, \text{ST}\}$.
Output: A maximum D-TOB $\mathcal{T} = (V_{\mathcal{T}}, A_{\mathcal{T}}, \tau_{\mathcal{T}})$ of $\mathcal{G}$ with root r.

1 $\mathcal{EA}(r) \leftarrow 0; \forall v \in V \setminus \{r\}, \mathcal{EA}(v) \leftarrow +\infty;$
2 $d(r) \leftarrow 0; \forall v \in V \setminus \{r\}, d(v) \leftarrow \text{D}_{\mathcal{G}}(r, v);$
3 $\langle d_1, \ldots, d_h \rangle \leftarrow$ ordered list of finite d values with no repetitions;
4 $V_{\mathcal{T}} \leftarrow \{r\}$; $A_{\mathcal{T}} \leftarrow \emptyset$; $\tau_{\mathcal{T}} \leftarrow 0$, $D_0 \leftarrow \{r\}$;
5 **for** $i = 1, \ldots, h$ **do**
6     $D_i \leftarrow \{v \in V \setminus \{r\} : d(v) = d_i\}$;
7   **if** $\text{D} = \text{LD}$ **then** enqueue all $(r, v, s, t) \in A$ such that $s = d_i$ in a min priority queue Q with weight t;
8   **if** $\text{D} = \text{ST}$ **then** enqueue all $(u, v, s, t) \in A$ such that $u \in D_0 \cup \ldots \cup D_{i-1}$ and $v \in D_i$ in a min priority queue Q with weight t;
9   **while** $Q \neq \emptyset$ **do**
10   dequeue $a \leftarrow (u, v, s, t)$ from Q;
11   **while** $s < \mathcal{EA}(u)$ *or* $t \geq \mathcal{EA}(v)$ *or* ($\text{D} = \text{ST}$ *and* $t - s \neq d_i - d(u)$) **do**
12   **if** $Q = \emptyset$ **then** go to Line 5 with next value of i;
13   dequeue $a \leftarrow (u, v, s, t)$ from Q;
14   **end**
        /* $a = (u, v, s, t)$ is s.t. $a \in \arg\min_{(u', v', s', t') \in Q} t'$, $s \geq \mathcal{EA}(u)$, $t < \mathcal{EA}(v) = +\infty$, and if $\text{D} = \text{ST}$, $t - s = d_i - d(u)$. */
15         $\mathcal{EA}(v) \leftarrow t$, $V_{\mathcal{T}} \leftarrow V_{\mathcal{T}} \cup \{v\}$, $A_{\mathcal{T}} \leftarrow A_{\mathcal{T}} \cup \{a\}$, $\tau_{\mathcal{T}} \leftarrow \max\{\tau_{\mathcal{T}}, t\}$;
16   enqueue all $(v, v', s', t') \in A$ such that $v' \in D_i$ in Q with weight t';
17   **end**
18 **end**

time. These ideas are formalized below. At step i of the **for** loop at lines 5-18, Algorithm 2 adds to $\mathcal{T}$ the vertices of D_i that are reachable by a D-prefix-optimal walk. To this aim, it uses a min priority queue Q for temporal arcs a with head vertices in D_i with weight $t_a(a)$. For $\text{D} = \text{LD}$, Q is initialized with all the outgoing temporal arcs from r with starting time d_i, as they are the only arcs that can realize a latest departure time equal to d_i. For $\text{D} = \text{ST}$, Q is initialized with all the temporal arcs with tail in $D_0 \cup \ldots \cup D_{i-1}$ and head in D_i. The vector $\mathcal{EA}$ in the algorithm, initialized at $+\infty$ for all the vertices but the root, keeps track of the arrival time in the vertices every time they are added to the TOB. In the **while** loop at lines 9-17, we dequeue temporal arcs from Q that cannot possibly be within an EAD-prefix-optimal walk. Formally, if such loop is not broken in line 12, then at the end we are left with an arc $a = (u, v, s, t) \in \arg\min_{(u', v', s', t') \in Q} t'$, i.e. an arc that minimizes the arrival time in the queue, satisfying:

- $s \geq \mathcal{EA}(u)$, so that a is temporally compatible with the temporal (r, u)-walk W_u that is already present in the TOB, i.e. $W_u + (u, a, v)$ is a temporal walk;
- $t < \mathcal{EA}(v)$, which ensures that we add to the TOB a new vertex each time;
- $t - s = d_i - d(u)$ if $\text{D} = \text{ST}$, ensuring that $W_u + (u, a, v)$ realizes $\text{ST}(r, v)$.

We then add v and the temporal arc a to the TOB and we update the arrival time in v to $\mathcal{EA}(v) = t$, which is equal to $\text{EAD}(r, v)$ and will be no longer updated until the end of the algorithm. Finally, we add to Q all the outgoing arcs from v with head vertices in D_i. When at distance d_i there are no arcs satisfying these constraints, i.e. the queue Q at line 12 is empty, we go to the next distance d_{i+1}, as it means that we have already spanned all the possible vertices in D_i. The initial computation of all $\textsc{d}(r, v)$ requires $O(M \log M)$ by Table 1. Concerning the remaining part of the algorithm, the i-th iteration of the **for** loop considers only arcs whose head is in D_i, hence each arc is considered only in one of the iterations of the **for** loop. Moreover, each arc is dequeued from Q at most once. As the dequeue from Q costs $O(\log M)$ we obtain a running time of $O(M \log M)$.

Theorem 3. *For any* $\textsc{d} \in \{\textsc{ld}, \textsc{st}\}$*, Algorithm 2 returns a maximum* D-TOB *of a temporal graph, for a chosen root, in* $O(M \log M)$ *time. Besides, the output is an* EAD-TOB.

Proof. See Appendix H.

5 Computing maximum FT-TOBs

As previously observed, Theorem 1 does not hold for $\textsc{d} = \textsc{ft}$. Indeed for FT the problem becomes NP-complete even in the following very constrained situations: when $el(a) = 0$ for all $a \in A$, also called *nonstrict* temporal graphs, and when $el(a) = 1$ for all $a \in A$, also called *strict* temporal graphs (see e.g. [5]). The nonstrict model is used when the time-scale of the measured phenomenon is relatively big: this is the case in a disease-spreading scenario [26] where the spreading speed might be unclear, or in time-varying graphs [21], where a single snapshot corresponds e.g. to all the streets available within a day.[8]

Theorem 4. *Let* $\mathcal{G} = (V, A, \tau)$ *be a temporal graph and* $r \in V$*. Deciding whether* $\mathcal{G}$ *admits a spanning* FT-TOB *with root* r *is* NP*-complete, even if* $\tau = 2$ *and* $el(a) = 0$ *for every* $a \in A$*, or if* $\tau = 3$ *and* $el(a) = 1$ *for every* $a \in A$*.*

Proof. See Appendix I.

The gaps left by the above theorem are when $\mathcal{G}$ has lifetime 1 or when $\mathcal{G}$ has lifetime 2 and all arcs have elapsed time at least 1. In the first case, the temporal graph reduces to a static graph, so the problem is solvable in polynomial time by Dijkstra's algorithm. As for when $\mathcal{G}$ has lifetime 2 and all arcs have elapsed time at least 1, one can see that the maximum FT-TOB rooted in r contains exactly r and every $u \in V$ such that $(r, u, 1, 2)$ is an arc in $\mathcal{G}$.

[8] The literature often focused on nonstrict/strict variations to provide stronger negative results. In this paper, we have used the more general model to provide stronger positive results, while using the nonstrict/strict when providing negative ones.

6 Conclusions and future work

We have showed that for D $\in$ {MT,ST,LD}, a spanning D-TOB does not always exist, but computing a D-TOB that spans the maximal number of vertices is polynomial-time. When D=FT, also finding a maximum FT-TOB becomes NP-complete. The fact that not all the vertices can be spanned by a maximum D-TOB could be an issue, for example, in a public transports setting, where we still want to reach all possible places. A natural follow-up of our work would be to relax the definition of spanning D-TOB, by asking to find a subgraph that reaches all the vertices from the root with a path realizing the distance, while having the least amount of arcs possible. Preliminary results suggest that this might become a much harder problem.

References

1. Bentert, M., Himmel, A., Nichterlein, A., Niedermeier, R.: Efficient computation of optimal temporal walks under waiting-time constraints. App. Net. Sci. **5** (2020)
2. Brunelli, F., Viennot, L.: Minimum-cost temporal walks under waiting-time constraints in linear time. arXiv:2211.12136 (2023)
3. Calamai, M., Crescenzi, P., Marino, A.: On computing the diameter of (weighted) link streams. ACM J. Exp. Algorithmics **27**, 4.3:1–4.3:28 (2022)
4. Campos, V.A., Lopes, R., Marino, A., Silva, A.: Edge-disjoint branchings in temporal graphs. Electronic J. of Combinatorics **28** (2020)
5. Casteigts, A.: Finding structure in dynamic networks. arXiv:1807.07801 (2018)
6. Cook, S.: The complexity of theorem-proving procedures. In: Proceedings of the third annual ACM symposium on Theory of computing. pp. 151–158 (1971)
7. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. McGraw–Hill, MIT Press, third ed. edn. (2001)
8. Dibbelt, J., Pajor, T., Strasser, B., Wagner, D.: Connection scan algorithm. ACM J. Exp. Algorithmics **23** (2018)
9. Gabow, H.N., Galil, Z., Spencer, T., Tarjan, R.: Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. Combinatorica **6**, 109–122 (1986)
10. Gallier, J.: Discrete Mathematics. Universitext, Springer (2011)
11. Gunturi, V., Shekhar, S., Bhattacharya, A.: Minimum spanning tree on spatio-temporal networks. In: Database and Expert Systems Applications. pp. 149–158. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
12. Holme, P., Saramäki, J.: Temporal networks. Phys. Reports **519**(3), 97–125 (2012)
13. Huang, S., Fu, A.W.C., Liu, R.: Minimum spanning trees in temporal graphs. In: ACM SIGMOD Intern. Conference on Management of Data. pp. 419–430 (2015)
14. Kamiyama, N., Kawase, Y.: On packing arborescences in temporal networks. Information Processing Letters **115**(2), 321–325 (2015)
15. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. Journal of Computer and System Sciences **64**(4), 820–842 (2002)
16. Kuwata, Y., Blackmore, L., Wolf, M., Fathpour, N., Newman, C., Elfes, A.: Decomposition algorithm for global reachability analysis on a time-varying graph with an application to planetary exploration. Intelligent Robot and Syst. (2009), 3955-3960
17. Latapy, M., Viard, T., Magnien, C.: Stream Graphs and Link Streams for the Modeling of Interactions over Time. Social Network Analysis **8**(1), 61:1–61:29 (2018)

18. Levin, L.: Universal sequential search problems. Problemy peredachi informatsii **9**(3), 115–116 (1973)
19. Marino, A., Silva, A.: Eulerian walks in temporal graphs. Algorithmica **85**, 805–830 (2023)
20. Michail, O.: An introduction to temporal graphs: An algorithmic perspective. Internet Mathematics **12**(4) (7 2016)
21. Nicosia, V., Tang, J., Musolesi, M., Russo, G., Mascolo, C., Latora, V.: Components in time-varying graphs. Chaos: An Interdisciplinary Journal of Nonlinear Science **22**(2) (2012)
22. Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., Samatova, N.: Anomaly detection in dynamic networks: a survey. WIREs Computational Statistics **7**(3), 223–247 (2015)
23. Tang, J.K., Mascolo, C., Musolesi, M., Latora, V.: Exploiting temporal complex network metrics in mobile malware containment. 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks pp. 1–9 (2010)
24. Wu, H., Cheng, J., Huang, S., Ke, Y., Lu, Y., Xu, Y.: Path problems in temporal graphs. Proc. VLDB Endow. **7**(9), 721–732 (2014)
25. Wu, H., Cheng, J., Ke, Y., Huang, S., Huang, Y., Wu, H.: Efficient algorithms for temporal path computation. Knowledge and Data Eng. **28**(11), 2927–2942 (2016)
26. Zschoche, P., Fluschnik, T., Molter, H., Niedermeier, R.: The complexity of finding small separators in temporal graphs. J. of Comp. and Syst. Sci. **107**, 72–92 (2020)

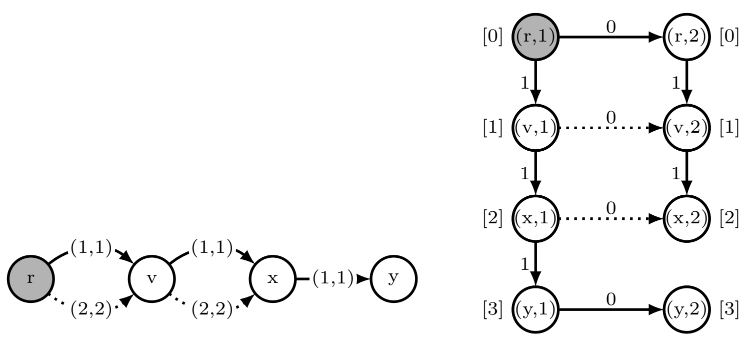
A The Static Expansion is not enough to compute a D-TOB

The *static expansion* of a temporal graph $\mathcal{G} = (V, A, \tau)$ is defined as the digraph $SE(\mathcal{G}) = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V} = \{(v,t) : v \in V,\, t \in [\tau]\}$ and $\mathcal{A} = \mathcal{M} \cup \mathcal{W}$ where $\mathcal{M} = \{((u,s),(v,t)) : (u,v,s,t) \in A\}$ and $\mathcal{W} = \{((v,t),(v,t+1)) : v \in V, t \in [\tau-1]\}$. We call $\mathcal{M}$ the set of *moving* arcs and $\mathcal{W}$ the set of *waiting* arcs. This graph has already being introduced in the literature, e.g. in [17, 20]. Some expediences can also be introduced in its definition in order to make its dimension being linear in the size of the original temporal graph, without altering its properties. The static expansion can be used for computing single source distances $\mathrm{D}_{\mathcal{G}}(r,v)$ by using Dijkstra's algorithm on it, given that each arc in $\mathcal{A}$ is provided by a suitable weight. We recall that in general, given a weighted directed graph $\mathcal{D} = (V', A')$ with non-negative weights, the Dijkstra's algorithm can compute the shortest paths from a given vertex $r \in V'$ (the *source* vertex) to all the others, producing a shortest-path out-branching with root r, with running time $O((|A'| + |V'|) \log |V'|)$, which reduces to $O(|A'| \log |V'|)$ if all vertices are reachable from the source. In particular by running Dijkstra's algorithm on $SE(\mathcal{G})$ with source $(r,1)$ we obtain:

- $\mathrm{EA}_{\mathcal{G}}(r,v)$ if we give a weight of 1 to all $a \in \mathcal{W}$ and a weight of $(t-s)$ to all $(u,v,s,t) \in \mathcal{M}$;
- $\mathrm{MT}_{\mathcal{G}}(r,v)$ if we give a weight of 0 to all $a \in \mathcal{W}$ and a weight of 1 to all $a \in \mathcal{M}$;
- $\mathrm{ST}_{\mathcal{G}}(r,v)$ if we give a weight of 0 to all $a \in \mathcal{W}$ and a weight of $(t-s)$ to all $(u,v,s,t) \in \mathcal{M}$;

- $\text{LD}_{\mathcal{G}}(r, v)$ by computing $\text{EA}(v, r)$ on $\mathcal{G}^{\circlearrowleft}$ (see Section 3.2);
- $\text{FT}_{\mathcal{G}}(r, v)$ if we give a weight of 0 to all $\{(r, t), (r, t+1)) : t \in [\tau - 1]\}$, a weight 1 to all the other waiting arcs, while every $(u, v, s, t) \in \mathcal{M}$ gets a weight of $(t - s)$. In this case, $\text{FT}_{\mathcal{G}}(r, v)$ is the minimum value among the ones obtained for the vertices $\{(v, t) : t \in [\tau]\}$.

Unfortunately the out-branching that the Dijkstra's algorithm returns on $SE(\mathcal{G})$ does not translate into a D-TOB of $\mathcal{G}$. The first problem is that by collapsing back all the vertices $\{(v, t) : t \geq 1\}$ to v, it is not guaranteed that the indegree of every vertex will remain equal to 1. The point is that we have to choose only one arc incoming to the group of temporal vertices (v, t) corresponding a same vertex v of the temporal graph. For example, consider the temporal graph of Figure (4a) and the out-branching produced by Dijkstra's algorithm on its static expansion in Figure (4b): if we collapse again the vertices of the out-branching, we get as a result the original temporal graph itself. Moreover, notice that in the example the vertices $(v, 1)$ and $(v, 2)$ reach the same distance through the Dijkstra's algorithm (the final distances are put in square brackets next to the vertices), and the same happens to the vertices $(x, 1)$ and $(x, 2)$. But only the choice of $(v, 1)$ and $(x, 1)$ will let us achieve a maximum MT-TOB, while the choice of $(v, 2)$ and $(x, 2)$ will not. A similar example can be produced for the other distances.



(a) Temporal graph $\mathcal{G}$ that admits an MT-TOB with root r (solid arcs)

(b) Static expansion of $\mathcal{G}$ and out-branching as output of Dijkstra's algorithm (solid arcs). Numbers in square bracket represent the MT distance computed.

Fig. 4: Example when the Dijkstra's algorithm on the static expansion present parities between vertices but only one choice leads to a TOB.

B Proof of Lemma 1

Lemma 1 *Let $\mathcal{T} = (V, A, \tau)$ be a temporal graph. The following are equivalent:*

1. *$\mathcal{T}$ is a* TOB *with root r;*

2. *For all $v \in V$ there is a temporal (r,v)-walk in $\mathcal{T}$. Additionally, $d^-_{\mathcal{T}}(r) = 0$ and, for all $v \in V \setminus \{r\}$, $d^-_{\mathcal{T}}(v) = 1$;*
3. *For all $v \in V$ there is a temporal (r,v)-walk in $\mathcal{T}$, and $|A| = |V| - 1$;*
4. *The underlying digraph $\mathcal{D}_{\mathcal{T}}$ of $\mathcal{T}$ is an out-branching with root r and for all $v \in V$, the unique (r,v)-walk in $\mathcal{T}$ is temporal.*

Proof. 1. $\implies$ 2. Since the existence of a temporal (r,v)-walk in $\mathcal{T}$ for all $v \in V$ is guaranteed by definition, we just need to show that $d^-_{\mathcal{T}}(r) = 0$ and that, for every $v \in V \setminus \{r\}$, $d^-_{\mathcal{T}}(v) = 1$. To that purpose, we first describe the set A of arcs of $\mathcal{T}$. Since from every temporal walk it is possible to extract a temporal path with the same extremes, we have that there exists a (r,v)-path in $\mathcal{T}$ for all $v \in V$. Fix now one (r,v)-path P_v for each $v \in V$. By the minimality of A, we deduce that

$$A = \bigcup_{v \in V} A(P_v). \tag{2}$$

As an immediate consequence of (2), there exists no arc in A entering in r and hence $d^-_{\mathcal{T}}(r) = 0$. Now suppose, by contradiction, that there exists $v \in V \setminus \{r\}$ such that $d^-_{\mathcal{T}}(v) \neq 1$. If $d^-_{\mathcal{T}}(v) = 0$, then v is not reachable from r, a contradiction. Thus we must have $d^-_{\mathcal{T}}(v) \geq 2$. Let $a_1, a_2 \in A$ be two different incoming arcs of v with $t_a(a_1) \leq t_a(a_2)$. We claim that we can delete the temporal arc a_2 from A while maintaining the property that every vertex is temporally reachable from r, and thus contradicting the minimality of A. Delete a_2. By (2), there exists $v_1 \in V$ such that $a_1 \in A(P_{v_1})$. Since in a path there cannot be two different arcs entering the same vertex, we have that $a_2 \notin A(P_{v_1})$, because $a_2 \neq a_1$. In particular, a_2 is not an arc for the v-prefix X of P_{v_1}. Let $u \in V$ and consider P_u. Assume that $a_2 \in P_u$. Then, since in a path an arc appears at most once, we have that a_2 does not appear in the v-suffix S of P_u. We consider then the (r,u)-walk given by $P = X + S$. Note that $a_2 \notin A(P)$ and that P is temporal because $t_a(a_1) \leq t_a(a_2)$. Hence u is temporally reachable from r after the removal of a_2, contradicting the minimality of A.
2. $\iff$ 3. $\iff$ 4. The temporal reachability from the root is guaranteed in each of the items. The other conditions are equivalent by the following lemma.

Lemma 2 ([10], Theorem 4.3). *Let $\mathcal{D} = (V, A)$ be a digraph and $r \in V$. The following facts are equivalent:*

1. *$\mathcal{D}$ is an out-branching with root r;*
2. *$\forall v \in V$ there exists a walk from r to v, $d^-_{\mathcal{D}}(r) = 0$, and $\forall v \in V \setminus \{r\}$, $d^-_{\mathcal{D}}(v) = 1$;*
3. *$\forall v \in V$ there exists a walk from r to v and $|A| = |V| - 1$.*

4. $\implies$ 1. The temporal reachability from the root is guaranteed by hypothesis. Since $\mathcal{D}_{\mathcal{T}}$ is an out-branching, by Lemma 2 it has $|V|-1$ arcs, which means that if we delete any arc then we necessarily disconnect some vertex from the root. Hence A is minimal.

C Proof of Proposition 1

Proposition 1 *Given a temporal graph $\mathcal{G}$, it holds that:*

1. *$\mathcal{T}$ is a maximum* EA-TIB *of $\mathcal{G}$ iff $\mathcal{T}^{\circlearrowleft}$ is a maximum* LD-TOB *of $\mathcal{G}^{\circlearrowleft}$;*
2. *$\mathcal{T}$ is a maximum* LD-TIB *of $\mathcal{G}$ iff $\mathcal{T}^{\circlearrowleft}$ is a maximum* EA-TOB *of $\mathcal{G}^{\circlearrowleft}$;*
3. *For each* $\textsc{d} \in \{\textsc{mt}, \textsc{st}, \textsc{ft}\}$*, $\mathcal{T}$ is a maximum* D-TIB *of $\mathcal{G}$ iff $\mathcal{T}^{\circlearrowleft}$ is a maximum* D-TOB *of $\mathcal{G}^{\circlearrowleft}$.*

Proof. Observe that $(\mathcal{G}^{\circlearrowleft})^{\circlearrowleft} = \mathcal{G}$ and that $W = (u, a_1, v_2, a_2, \ldots, v_k, a_k, v)$ is a temporal (u, v)-walk in $\mathcal{G}$ if and only if $W^{\circlearrowleft} = (v, a_k^{\circlearrowleft}, v_k, \ldots, a_2^{\circlearrowleft}, v_2, a_1^{\circlearrowleft}, u)$ is a temporal (v, u)-walk in $\mathcal{G}^{\circlearrowleft}$. Then note that for any walk W in $\mathcal{G}$ it holds that $(W^{\circlearrowleft})^{\circlearrowleft} = W$, $t_s(W^{\circlearrowleft}) = \tau - t_a(W) + 1$, $t_a(W^{\circlearrowleft}) = \tau - t_s(W) + 1$, $\ell(W) = \ell(W^{\circlearrowleft})$, $\mathrm{dur}(W) = \mathrm{dur}(W^{\circlearrowleft})$ and $\mathrm{tt}(W) = \mathrm{tt}(W^{\circlearrowleft})$. It is also easy to see that $\mathcal{T}$ is a TIB with root r if and only if $\mathcal{T}^{\circlearrowleft}$ is a TOB with root r. Let now W and W' be two (v, r)-walks in $\mathcal{G}$. We claim that W realizes $\textsc{ea}(v, r)$ in $\mathcal{G}$ if and only if $W^{\circlearrowleft}$ realizes $\textsc{ld}(r, v)$ in $\mathcal{G}^{\circlearrowleft}$. Indeed, $t_a(W) \leq t_a(W')$ if and only if $\tau - t_s(W^{\circlearrowleft}) + 1 \leq \tau - t_s(W'^{\circlearrowleft}) + 1$ if and only if $t_s(W^{\circlearrowleft}) \geq t_s(W'^{\circlearrowleft})$. Similarly, we claim that W realizes $\textsc{ld}(v, r)$ in $\mathcal{G}$ if and only if $W^{\circlearrowleft}$ realizes $\textsc{ea}(r, v)$ in $\mathcal{G}^{\circlearrowleft}$. Indeed, $t_s(W) \geq t_s(W')$ if and only if $\tau - t_a(W^{\circlearrowleft}) + 1 \geq \tau - t_a(W'^{\circlearrowleft}) + 1$ if and only if $t_a(W^{\circlearrowleft}) \leq t_a(W'^{\circlearrowleft})$. We now prove that, for $\textsc{d} \in \{\textsc{ft}, \textsc{mt}, \textsc{st}\}$, W realizes $\textsc{d}(v, r)$ in $\mathcal{G}$ if and only if $W^{\circlearrowleft}$ realizes $\textsc{d}(r, v)$ in $\mathcal{G}^{\circlearrowleft}$. In fact it holds that $\ell(W) \leq \ell(W')$ if and only if $\ell(W^{\circlearrowleft}) \leq \ell(W'^{\circlearrowleft})$, $\mathrm{tt}(W) \leq \mathrm{tt}(W')$ if and only if $\mathrm{tt}(W^{\circlearrowleft}) \leq \mathrm{tt}(W'^{\circlearrowleft})$ and $\mathrm{dur}(W) \leq \mathrm{dur}(W')$ if and only if $\mathrm{dur}(W^{\circlearrowleft}) \leq \mathrm{dur}(W'^{\circlearrowleft})$. This concludes our proof.

D Proof of Theorem 1

To prove Theorem 1, we first need the following Lemma.

Lemma 3. *Let W be a* D*-prefix-optimal temporal (r, u)-walk in $\mathcal{G}$ and $v \in V(W)$. Let S be a v-suffix of W and let W_v be a* D*-prefix-optimal temporal (r, v)-walk in $\mathcal{G}$. If $t_a(W_v) \leq t_s(S)$, then $W_v + S$ is a* D*-prefix-optimal temporal (r, u)-walk in $\mathcal{G}$.*

Proof. Since $t_a(W_v) \leq t_s(S)$ by hypothesis, then $\bar{W} = W_v + S$ is a temporal (r, u)-walk. Clearly $\bar{W}$ realizes $\textsc{d}_{\mathcal{G}}(r, x)$ for every $x \in V(W_v)$ by construction. We now need to prove that $\bar{W}$ realizes $\textsc{d}_{\mathcal{G}}(r, x)$ for all $x \in V(S)$. So consider an arbitrary $x \in V(S)$. If $\textsc{d} = \textsc{ld}$, then since W and W_v are LD-prefix-optimal, we get $t_s(W) = t_s(W_v) = \textsc{ld}(r, v) = \textsc{ld}(r, x)$, so we are done. Let now X be the v-prefix of W such that $W = X + S$. If $\textsc{d} = \textsc{mt}$, then W and W_v are paths and $\ell(X) = \textsc{mt}(r, v) = \ell(W_v)$. Consequently $\bar{W}$ realizes $\textsc{mt}(r, x)$. If $\textsc{d} = \textsc{st}$, then $\mathrm{tt}(X) = \mathrm{tt}(W_v)$. Consequently $\bar{W}$ realizes $\textsc{st}(r, x)$.

Theorem 1 *Let $\mathcal{G} = (V, A, \tau)$ be a temporal graph, $r \in V$ and $\textsc{d} \in \{\textsc{ld}, \textsc{mt}, \textsc{st}\}$. Then $\mathcal{G}$ admits a spanning* D-TOB *with root r if and only if for all $v \in V$, there exists a* D*-prefix-optimal temporal (r, v)-path in $\mathcal{G}$.*

Proof. $\Longrightarrow$ By the definition of D-TOB with root r and the uniqueness of temporal walks from the root to any other vertex in a TOB (see Lemma 1), it follows that all walks in a D-TOB are paths and are D-prefix-optimal.
$\Longleftarrow$ For every $v \in V$, let W_v be a temporal D-prefix-optimal (r, v)-path in $\mathcal{G}$. Let $A' := \bigcup_{v \in V} A(W_v)$. For $B \subseteq A'$, denote by $\mathcal{T}[B]$ the temporal subgraph of $\mathcal{G}$ having vertex set V and temporal arc set B, and consider the property:
$\mathcal{P}_{\text{D},B}$) for all $v \in V$, there exists in $\mathcal{T}[B]$ a temporal D-prefix-optimal (r, v)-walk. Note that $\mathcal{P}_{\text{D},A'}$ is satisfied. Thus it is possible to consider the minimal subsets B of arcs in A' satisfying $\mathcal{P}_{\text{D},B}$. Let $A_\mathcal{T} \subseteq A'$ be one of such minimal sets and let $\mathcal{T} := \mathcal{T}[A_\mathcal{T}]$. We show that $\mathcal{T}$ is a D-TOB for $\mathcal{G}$. Clearly, by the construction of $\mathcal{T}$, it is enough to show that $\mathcal{T}$ is a TOB. Furthermore, in view of Lemma 1, it suffices to show that $d^-_\mathcal{T}(r) = 0$ and that for all $v \in V \setminus \{r\}$, $d^-_\mathcal{T}(v) = 1$, since the temporal reachability from vertex r to any other vertex is already guaranteed by property $\mathcal{P}_{\text{D},A_\mathcal{T}}$. Since A' does not contain arcs entering in r, this holds also for $A_\mathcal{T}$ and hence we have that $d^-_\mathcal{T}(r) = 0$. Suppose now, by contradiction, that there exists $v \in V \setminus \{r\}$ such that $d^-_\mathcal{T}(v) \neq 1$. Since $d^-_\mathcal{T}(v) = 0$ implies that v is not reachable from r, we necessarily have $d^-_\mathcal{T}(v) \geq 2$. Let $a_1, a_2 \in A_\mathcal{T}$ be two different incoming temporal arcs of v with $t_a(a_1) \leq t_a(a_2)$. We claim that $\mathcal{P}_{\text{D},A_\mathcal{T} \setminus \{a_2\}}$ is satisfied, and thus the minimality of $A_\mathcal{T}$ is contradicted. Indeed, by definition of $A_\mathcal{T}$, there exists $v_1 \in V$ and a temporal (r, v_1)-path W_{v_1} such that $a_1 \in A(W_{v_1})$. Since in a path two distinct arcs entering in the same vertex do not appear, we have that $a_2 \notin A(W_{v_1})$. In particular, a_2 is not an arc for the v-prefix X of W_{v_1}. Let $u \in V$ and consider W_u a temporal (r, u)-path in $\mathcal{T}$. Assume that $a_2 \in W_u$. Then, since in a path an arc appears at most once, we have that a_2 does not appear in the v-suffix S of W_u. We consider then the (r, u)-walk given by $\bar{W} = X + S$. Then $a_2 \notin A(\bar{W})$ and we have that $t_a(X) = t_a(a_1) \leq t_a(a_2) \leq t_s(S)$. As a consequence, by Lemma 3, $\bar{W}$ is a D-prefix-optimal walk in $\mathcal{T}[A_\mathcal{T} \setminus \{a_2\}]$.

E Proof of Corollary 1

Corollary 1 *Let $\mathcal{G} = (V, A, \tau)$ be a temporal graph, $r \in V$, and* D $\in \{$LD, MT, ST$\}$*. Then a maximum* D-TOB *$\mathcal{T}$ with root r of $\mathcal{G}$ has vertex set:*

$$V_\mathcal{T} = \{v \in V : \textit{there exists a } \text{D}\textit{-prefix-optimal } (r, v)\textit{-walk in } \mathcal{G}\ \}$$

Proof. Consider $\mathcal{G}[V_\mathcal{T}]$. Let $v \in V_\mathcal{T}$ and W a D-prefix-optimal (r, v)-temporal walk in $\mathcal{G}$. By definition of D-prefix-optimal walk, for every $u \in V(W)$, it holds that $u \in V_\mathcal{T}$, which implies that W is also a D-prefix-optimal (r, v)-temporal walk in $\mathcal{G}[V_\mathcal{T}]$. Hence, by Theorem 1, $\mathcal{G}[V_\mathcal{T}]$ admits a spanning D-TOB $\mathcal{T}$, which is also a D-TOB of $\mathcal{G}$. We now show that $\mathcal{T}$ is maximum. By Remark 1 it suffices to prove that if $V' \subseteq V$ is such that $V' \setminus V_\mathcal{T} \neq \emptyset$, then $\mathcal{G}[V']$ does not admit a spanning D-TOB with root r. Let $u \in V' \setminus V_\mathcal{T}$: by hypothesis there does not exist a D-prefix-optimal temporal (r, u)-walk in $\mathcal{G}$, hence there does not exist one in $\mathcal{G}[V']$. By Theorem 1, $\mathcal{G}[V']$ does not admit a spanning D-TOB.

F Existence of ST-TOB when $\tau \leq 2$

Claim. If $\tau \leq 2$, then a temporal graph $\mathcal{G} = (V, A, \tau)$ admits a ST-TOB with root $r \in V$ if and only if each vertex is temporally reachable from r.

Proof. If $\tau = 1$ than the temporal graph reduces to a static digraph and the problem of finding a ST-TOB reduces to finding an out-branching, so the thesis follows. If $\tau = 2$, notice that every temporal label $(t_s(a), t_a(a))$ can assume only three values, namely $\{(1,1),(1,2),(2,2)\}$. This implies that every temporal walk W in $\mathcal{G}$ is such that either $\mathrm{tt}(W) = 0$ or $\mathrm{tt}(W) = 1$. Moreover, if $\mathrm{tt}(W) = 0$, then W is necessarily ST-prefix-optimal. We now want to prove that if v is temporally reachable from r, then there exists a ST-prefix-optimal (r, v)-walk in $\mathcal{G}$. Let W be a (r, v)-walk in $\mathcal{G}$. If $\mathrm{tt}(W_v) = 0$, we are done. Suppose now that $\mathrm{tt}(W) = 1$ and that W is not ST-prefix-optimal. Let u be the first vertex of W starting from v for which W does not realize $\mathrm{ST}(r, u)$. This means that there must exists a u-prefix X of W such that $\mathrm{tt}(X) = 1$ while $\mathrm{ST}(r, u) = 0$. This implies that $t_a(X) = 2$ and that there must exists a (r, u)-walk W_u in $\mathcal{G}$ such that $\mathrm{tt}(W_u) = 0$, so W_u is ST-prefix-optimal. Let S a u-suffix of W such that $W = X + S$; notice that $t_a(W_u) \leq 2 = t_a(X) = t_s(S)$. Then by construction $W_u + S$ is a ST-prefix-optimal (r, v)-walk in $\mathcal{G}$.

G Proof of Theorem 2

Theorem 2 *Algorithm 1 returns a maximum* MT-TOB *of a temporal, for a chosen root, graph in $O(M \log n)$ time. Besides, the output is an* EAMT-TOB.

Proof. Let $\mathcal{G} = (V, A, \tau)$ the temporal graph input of the algorithm and $r \in V$, $d(v) = \mathrm{MT}_{\mathcal{G}}(r, v)$ for all $v \in V$, $h = \max\{d(v) : v \in V\}$ and $V' = \{v \in V : v \text{ is temporally reachable from } r\}$. For $i \in [h]_0$ let $D_i = \{v \in V : d(v) = i\}$ and note that $\{D_i : i \in [h]_0\}$ is a partition of V' with $D_0 = \{r\}$. Since no confusion is possible, from now on we will avoid writing the subscripts $\mathcal{G}$. We prove the following loop invariant:

Claim. At the end of the i-th iteration of the **for** loop in lines 4-12, $V_{\mathcal{T}} = \{v \in V' : \exists \text{ an MT-prefix-optimal temporal } (r, v)\text{-walk in } \mathcal{G} \text{ and } d(v) \leq i\}$, $\mathcal{EA}(v) = \mathrm{EAMT}(r, v)$ for all $v \in V_{\mathcal{T}}$, $v \neq r$, and $\mathcal{T}$ is an EAMT-TOB with root r of $\mathcal{G}$.

The above claim implies that the final output $\mathcal{T}$ of the algorithm is an EAMT-TOB with root r of $\mathcal{G}$, which is in particular an MT-TOB with root r of $\mathcal{G}$. Moreover, $V_{\mathcal{T}}$ would consist of all the vertices in $\mathcal{G}$ for which there exists an MT-prefix-optimal temporal walk from the root, thus $\mathcal{T}$ is a maximum MT-TOB by Corollary 1. We are left to prove the claim. $\mathcal{T}$ is initialized as the temporal graph made of the sole vertex r, so the loop invariant is trivially true. Suppose now that the loop invariant is true up to a certain i-th iteration, we prove that it holds for the $(i+1)$-th iteration. Let $v \in V$ such that $d(v) = i + 1$. We first prove that if there exists an MT-prefix-optimal temporal (r, v)-walk in $\mathcal{G}$, say W, then the set S in line 6 is non-empty. We can always choose W such that it arrives the earliest in v among all the MT-prefix-optimal temporal (r,v)-walks, that is $t_a(W) = \text{EAMT}(r,v)$. Let $\bar{a} = (\bar{u}, v, \bar{s}, \bar{t}) \in A$ be the last temporal arc of W. It holds that $d(\bar{u}) = i$ and therefore, by the inductive hypothesis, we have that $\bar{u} \in V_i$ and $\mathcal{EA}(\bar{u}) = \text{EAMT}(r, \bar{u})$ at the end of the i-iteration. Since W is MT-prefix-optimal, it holds that $\bar{s} \geq \text{EAMT}(r,\bar{u}) = \mathcal{EA}(\bar{u})$. Therefore $\bar{a}$ fulfils the conditions to belong to S, i.e. S is non-empty. Notice also that since $\bar{a} \in S$ and W realizes $\text{EAMT}(r,v)$, we have that $\bar{t} = \text{EAMT}(r,v)$ and so

$$\min_{(u',v,s',t') \in S} t' = \bar{t} = \text{EAMT}(r,v)\,. \tag{3}$$

We now prove that $\mathcal{T}$ is an EAMT-TOB with root r. We have just showed that if there exists an MT-prefix-optimal temporal (r,v)-walk in $\mathcal{G}$, then S in line 6 is non-empty. This implies that in line 8 we choose an arc $a = (u,v,s,t) \in S$ that minimizes the arrival time, so this arc is added to $A_{\mathcal{T}}$, while v is added to $V_{\mathcal{T}}$ and $\mathcal{EA}(v)$ is set to t. Because $D_0, \ldots, D_h$ is a partition of V', no other incoming arc to v is added in the algorithm, and therefore v has in-degree equal to 1 in $\mathcal{T}$. Moreover $s \geq \mathcal{EA}(u)$ since $a \in S$, so if W_u is the unique temporal (r,u)-path in $\mathcal{T}$ (it exists by inductive hypothesis), then $W_v = W_u + (u,a,v)$ is a temporal (r,v)-path in $\mathcal{T}$. Hence $\mathcal{T}$ is a TOB with root r. It remains to show that W_v realizes $\text{EAMT}(r,v)$. By the inductive hypothesis we have that W_u is EAMT-prefix-optimal. Therefore $\ell(W_v) = \ell(W_u) + 1 = d(u) + 1 = i + 1 = d(v)$. Moreover, by equation (3) and since $a \in S$, we have that $\mathcal{EA}(v) = t_a(W_v) = t = \bar{t} = \text{EAMT}(r,v)$. This concludes the proof of claim.

Regarding the computational complexity of the algorithm, by Table 1 the initial computation of all distances requires $O(M \log n)$; the remaining part of the algorithm takes $O(M)$ as it requires only one scan of each temporal arc. Therefore the overall complexity is $O(M \log n)$.

H Proof of Theorem 3

Theorem 3 *For any* D $\in \{$LD, ST$\}$*, Algorithm 2 returns a maximum* D-TOB *of a temporal graph, for a chosen root, in $O(M \log M)$ time. Besides, the output is an* EAD-TOB.

Proof. Let $\mathcal{G} = (V, A, \tau)$ the temporal graph input of the algorithm and $r \in V$, $d(v) = \text{D}_{\mathcal{G}}(r,v)$ for all $v \in V$, $h = |\{d(v) : v \in V\}|$ and $\{d_0 < d_1 < \cdots < d_h\} = \{d(v) : v \in V\}$. Let $V' = \{v \in V : v$ is temporally reachable from $r\}$ and for all $i \in [h]_0$, let $D_i = \{v \in V : d(v) = d_i\}$. Note that $\{D_i : i \in [h]_0\}$ is a partition of V' with $D_0 = \{r\}$. Since no confusion is possible, from now on we will avoid writing the subscripts $\mathcal{G}$. Note that if D $=$ LD, then each iteration of the **for** loop in lines 5-18 is completely independent on the others, as it deals only with vertices in D_i and temporal arcs with both tail and head in D_i. We now proceed by proving the following loop invariant:

Claim. Given $\text{D} \in \{\text{LD}, \text{ST}\}$, at the end of the i-th iteration of the **for** loop in lines 5-18, we have that $\mathcal{EA}(v) = \text{EAD}_{\mathcal{G}}(r, v)$ for all $v \in V_{\mathcal{T}}$, $v \neq r$, and that $\mathcal{T} = (V_{\mathcal{T}}, A_{\mathcal{T}}, \tau_{\mathcal{T}})$ is an EAD-TOB with root r of $\mathcal{G}$ with

$$V_{\mathcal{T}} = \{v \in V' : \exists \text{ a D-prefix-optimal temp. } (r, v)\text{-walk in } \mathcal{G} \text{ and } d(v) \leq d_i\}. \quad (4)$$

The claim above implies that the final output $\mathcal{T}$ of the algorithm is an EAD-TOB with root r of $\mathcal{G}$, which is in particular a D-TOB with root r of $\mathcal{G}$. Moreover, $V_{\mathcal{T}}$ would consist of all the vertices in $\mathcal{G}$ for which there exists an D-prefix-optimal temporal walk from the root, thus $\mathcal{T}$ is a maximum D-TOB by Corollary 1. We are left to prove the claim. $\mathcal{T}$ is initialized as the temporal graph made of the sole vertex r, so the loop invariant is trivially true. Suppose now that the loop invariant is true up to a certain i-th iteration, we prove that it holds for the $(i+1)$-th iteration. We start by proving that if $v \in D_{i+1}$ and there exists a D-prefix-optimal temporal (r, v)-walk in $\mathcal{G}$, then $v \in V_{\mathcal{T}}$ at the end of the $(i+1)$-th **for** loop iteration. Let $W = (r = x_0, a_1, x_1, \ldots, a_m, x_m = v)$ be one of these D-prefix-optimal temporal (r, v)-walks in $\mathcal{G}$. Let x_j be the last vertex of W starting from r that is in $V_{\mathcal{T}}$ before the beginning of the $(i+1)$-th iteration (x_j possibly equal to r). By inductive hypothesis, this implies that $d(x_j) < d_{i+1}$ and that $d(x_l) = d_{i+1}$ for all $l > j$. Then the arc a_{j+1} is added to Q in lines 7-8 at the beginning of the $(i+1)$-th iteration. Since W is D-prefix-optimal, a_{j+1} does not fulfil the condition in line 11, unless x_{j+1} has been already added to $V_{\mathcal{T}}$. This implies that in any case at one point of the **while** loop x_{j+1} is being added to $V_{\mathcal{T}}$, which implies that a_{j+2} is put in queue Q by line 16. This iteratively proves that for all $l > j$, x_l will be added to $V_{\mathcal{T}}$ at one point of the **while** loop, and this includes $x_m = v$. This proves equation (4). To prove the rest of the claim, we are going to prove the following fact:

Claim. When in the $(i+1)$-th iteration of the **for** loop of lines 5-18, at the end of each iteration of the **while** loop of lines 9–17, we have that $\mathcal{T} = (V_{\mathcal{T}}, A_{\mathcal{T}}, \tau_{\mathcal{T}})$ is an EAD-TOB with root r and $\forall v \in V_{\mathcal{T}}$, $\mathcal{EA}(v) = \text{EAD}_{\mathcal{G}}(r, v)$.

At the beginning of the $(i+1)$-th **for** loop iteration, the inductive hypothesis holds, so the invariant property is true. By contradiction, consider the first iteration of the **while** loop such that the addition of the vertex v to $V_{\mathcal{T}}$ and of the arc $a = (u, v, s, t)$ to $A_{\mathcal{T}}$ makes the claim fail. Since we are at the $(i+1)$-th **for** loop iteration, it holds that $d(v) = d_{i+1}$. Due to line 11, it must hold that $s \geq \mathcal{EA}(u)$ and, if $\text{D} = \text{ST}$, then $t - s = d_{i+1} - d(u)$. This implies that $\mathcal{EA}(u) < +\infty$, and since the only way for this to hold is to have $u = r$, or to have $\mathcal{EA}(u)$ updated to a natural number, in which case u is added to $V_{\mathcal{T}}$ (line 15), we get that $u \in V_{\mathcal{T}}$. Also, u must have entered $V_{\mathcal{T}}$ before v, so by hypothesis there exists a (unique) EAD-prefix-optimal temporal (r, u)-walk W_u in $\mathcal{T}$; in particular $t_a(W_u) = \text{EAD}(r, u) = \mathcal{EA}(u)$. Since $s \geq \mathcal{EA}(u)$, then the walk $W_v = W_u + (u, a, v)$ is a temporal (r, v)-walk in $\mathcal{T}$. Moreover, if $\text{D} = \text{LD}$, then $u \in D_{i+1}$, so $t_s(W_v) = t_s(W_u) = d(u) = d_{i+1} = d(v)$. If $\text{D} = \text{ST}$, then $t - s = d_{i+1} - d(u)$, so $\text{tt}(W_v) = \text{tt}(W_u) + (t - s) = d(u) + (t - s) = d(u) + (t - s) = d_{i+1}$. Hence in both cases, W_v realizes $d(v)$ and it is D-prefix-optimal. This also implies that $t = t_a(W_v) \geq \text{EAD}(r, v)$. It remains to show that v has indegree 1 in $\mathcal{T}$ and that $t = \mathcal{EA}(v) = \textsc{ead}(r, v)$ to derive the contradiction. Suppose first that v has indegree $\neq 1$. Then it must have indegree > 1 since a is an incoming temporal arc of v that belongs to $A_{\mathcal{T}}$. Then at a previous step of the **while** loop an arc $a' = (u', v, s', t')$ was added to $A_{\mathcal{T}}$, which means that in the same step also v was added to $V_{\mathcal{T}}$ and $\mathcal{EA}(v)$ was set equal to t'. When a' was added, u' must have already been in $V_{\mathcal{T}}$. By hypothesis, there exists a (unique) $\textsc{ead}$-prefix-optimal temporal (r, u')-walk $W_{u'}$ in $\mathcal{T}$, and $W' = W_{u'} + (u', a', v)$ is such that $\textsc{ead}(r, v) = t_a(W') = t' = \mathcal{EA}(v)$ by hypothesis. Since $t \geq \textsc{ead}(r, v) = \mathcal{EA}(v)$, the arc a could have never been chosen later, as it is fulfilling the condition $t \geq \mathcal{EA}(v)$ in line 11. So v has indegree 1 in $\mathcal{T}$. Suppose now that $\mathcal{EA}(v) \neq \textsc{ead}(r, v)$, i.e. that $t > \textsc{ead}(r, v)$. We know that v has a $\textsc{d}$-prefix-optimal (r, v)-walk in $\mathcal{G}$; let W be the $\textsc{d}$-prefix-optimal (r, v)-walk in $\mathcal{G}$ that arrives the earliest in v, i.e. $t_a(W) = \textsc{ead}(r, v)$. Let $y \in V(W)$ be the first vertex along W such that, when v is added to $V_{\mathcal{T}}$, $y \notin V_{\mathcal{T}}$, and let $x \in V_{\mathcal{T}}$ be y's predecessor along W and $a_{xy} = (x, y, s_{xy}, t_{xy})$ the temporal arc connecting them in W (x may coincide with r). By inductive hypothesis we have that $y \in D_{i+1}$. Notice that $t_{xy} \leq t_a(W) = \textsc{ead}(r, v)$. Since $x \in V_{\mathcal{T}}$ and we chose v as the first vertex for which $\mathcal{EA}(v) \neq \textsc{ead}(r, v)$, we have that $\mathcal{EA}(x) = \textsc{ead}(r, x)$ when x was added to $V_{\mathcal{T}}$. This implies that the arc a_{xy} is enqueued in Q when x is added to $V_{\mathcal{T}}$. Indeed if $\textsc{d} = \textsc{ld}$, then $d(x) = d(y) = d(v) = d_{i+1}$ and so $a_{xy} \in \{(x, v', s', t') \in A : v' \in D_{i+1}\}$ in line 16. If $\textsc{d} = \textsc{st}$, let $d_{i'} = d(x) \leq d_{i+1}$. If $i' = i + 1$ we conclude as above. If $i' < i + 1$, since $y \in D_{i+1}$, then a_{xy} is enqueued in Q at the beginning of the $(i + 1)$-th **for** loop iteration (line 8). We claim that when v was added to $V_{\mathcal{T}}$, a_{xy} was still in Q. Indeed, a_{xy} could have not been dequeued from Q and added to $A_{\mathcal{T}}$ since otherwise $y \in V_{\mathcal{T}}$ before v was added to $V_{\mathcal{T}}$, which contradicts the hypothesis. If a_{xy} was dequeued from Q without being added to $A_{\mathcal{T}}$, since W is $\textsc{d}$-prefix-optimal (and so $t_{xy} - s_{xy} = d(y) - d(x) = d_{i+1} - d(x)$ if $\textsc{d} = \textsc{st}$) and $s_{xy} \geq \textsc{ead}(r, x) = \mathcal{EA}(x)$, then it must have hold that $\mathcal{EA}(y) < +\infty$. This implies that y was already in $V_{\mathcal{T}}$ before v was added to $V_{\mathcal{T}}$, which again contradicts the hypothesis. Therefore, when v was added to $V_{\mathcal{T}}$, it must hold that $t \leq t_{xy}$. But $t_{xy} \leq t_a(W) = \textsc{ead}(r, v) \leq t$ and so $t = \textsc{ead}(r, v)$. This concludes the proof. Regarding the computational complexity of the algorithm, the initial computation of all $\textsc{d}(r, v)$, $v \in V$, requires $O(M \log M)$ by Table 1. Concerning the remaining part of the algorithm, notice that the i-th iteration of the **for** loop considers only arcs whose head is in D_i. This means that each arc will be considered only in one of the iterations of the **for** loop. Moreover, each arc is dequeued from Q at most once. As the dequeue from Q costs $O(\log M)$ we obtain a total running time of $O(M \log M)$.

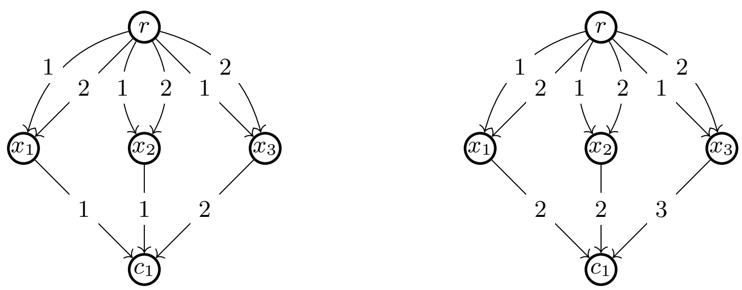
I Proof of Theorem 4

Theorem 4 *Let $\mathcal{G} = (V, A, \tau)$ be a temporal graph and $r \in V$. Deciding whether $\mathcal{G}$ admits a spanning* FT-TOB *with root r is* NP*-complete, even if $\tau = 2$ and $el(a) = 0$ for every $a \in A$, or if $\tau = 3$ and $el(a) = 1$ for every $a \in A$.*

Proof. The problem is in NP, since computing $\textsc{ft}_{\mathcal{G}}(r, v)$ for every vertex v can be done in polynomial time (Table 1), and because testing whether a given temporal subgraph $\mathcal{T}$ is a TOB can be done in polynomial time (see e.g. Lemma 1). To prove hardness, we make a reduction from 3-SAT, largely known to be NP-complete [6, 18]. For this, consider a formula ϕ in CNF form on variables $X = \{x_1, \ldots, x_n\}$ and on clauses $C = \{c_1, \ldots, c_m\}$. We first construct $\mathcal{G} = (V, A, \tau)$ for the case where every arc has elapsed time 0 (observe Figure (5a) to follow the construction). First, let $V = X \cup C \cup \{r\}$. For each variable x_i, add to A the temporal arcs $(r, x_i, 1, 1)$ and $(r, x_i, 2, 2)$. Then, for each clause c_j and each variable x_i appearing in c_j, add temporal arc $(x_i, c_j, 1, 1)$ if x_i appears in c_j positively, while add the temporal arc $(x_i, c_j, 2, 2)$ if x_i appears in c_j negatively. We now prove that ϕ is satisfiable if and only if there exists a spanning FT-TOB rooted in r. Suppose first that ϕ admits a satisfying assignment; we show how to construct a spanning FT-TOB $\mathcal{T} = (V, A_{\mathcal{T}}, \tau_{\mathcal{T}})$ rooted in r. For each variable x_i, add to $A_{\mathcal{T}}$ the temporal arc $(r, x_i, 1, 1)$ if x_i is true, while add to $A_{\mathcal{T}}$ the temporal arc $(r, x_i, 2, 2)$ if x_i is false. Now consider a clause c_j and choose one of the variables that validates c_j, say x_{i_j}. Add to $A_{\mathcal{T}}$ the unique temporal arc with head c_j and tail x_{i_j}. Now observe that the vertices in X are connected to r in $\mathcal{T}$ through direct arcs; hence we get that $\textsc{ft}_{\mathcal{T}}(r, x_i) = 0$ for every $x_i \in X$. For a clause c_j, if x_{i_j} appears positively in c_j, then x_{i_j} is true, and $(r, x_{i_j}, 1, 1)$ and $(x_{i_j}, c_j, 1, 1)$ are in $A_{\mathcal{T}}$; therefore $\textsc{ft}_{\mathcal{T}}(r, c_j) = 0$. If x_{i_j} appears negatively in c_j, then x_{i_j} is false, so $(r, x_{i_j}, 2, 2)$ and $(x_{i_j}, c_j, 2, 2)$ are in $A_{\mathcal{T}}$; therefore $\textsc{ft}_{\mathcal{T}}(r, c_j) = 0$. Finally, observe that each vertex different from the root has indegree 1. By Lemma 1, we get that $\mathcal{T}$ is a spanning TOB, and since $\textsc{ft}_{\mathcal{T}}(r, v) = 0$ for every $v \in V$, it follows that $\mathcal{T}$ is a spanning FT-TOB.

Suppose now that $\mathcal{T} = (V, A_{\mathcal{T}}, \tau_{\mathcal{T}})$ is a spanning FT-TOB rooted in r. Since the only possible (r, x_i)-walk is through an arc, we get that either $(r, x_i, 1, 1) \in A_{\mathcal{T}}$ or $(r, x_i, 2, 2) \in A_{\mathcal{T}}$. If the former occurs, then set x_i to true, while if the latter occurs, then set x_i to false. We now argue that this must be a satisfying assignment to ϕ. For this, consider a clause c_j. By Lemma 1, we know that $d^-_{\mathcal{T}}(c_j) = 1$; so let $a = (x_{i_j}, c_j, t, t)$ be the temporal arc incident to c_j in $\mathcal{T}$. If x_{i_j} appears positively in c_j, then we know that $a = (x_{i_j}, c_j, 1, 1)$ by construction. And since the temporal (r, c_j)-walk must pass by x_{i_j}, we get that $(r, x_{i_j}, 1, 1) \in A_{\tau}$, in which case x_{i_j} is set to true and hence satisfies c_j. If x_{i_j} appears in c_j negatively, then $a = (x_{i_j}, c_j, 2, 2)$. Notice that $\textsc{ft}_{\mathcal{G}}(r, c_j) = 0$; since $\mathcal{T}$ is an FT-TOB, we must also have $\textsc{ft}_{\mathcal{T}}(r, c_j) = 0$. This implies that $(r, x_{i_j}, 2, 2) \in A_{\mathcal{T}}$ and hence x_{i_j} is set to false, satisfying c_j.

In the case where $el(a) = 1$ for every arc a, the reduction is similar to the previous one. Specifically, for each $x_i \in X$, we add arcs $(r, x_i, 1, 2)$ and $(r, x_i, 2, 3)$. For each clause c_j, if x_i appears positively in c_j we add the temporal arc $(x_i, c_j, 2, 3)$, while if x_i appears negatively in c_j we add the temporal arc $(x_i, c_j, 3, 4)$. Analogous arguments to the previous ones apply.



(a) All temporal arcs have elapsed time 0. (b) All temporal arcs have elapsed time 1.

Fig. 5: Example of the construction in the proof of Theorem 4. Clause c_1 is equal to $(x_1 \vee x_2 \vee \neg x_3)$. The value on top of each arc represents the starting time.