HAL open science

Temporally connected components

Stefan Balev, Yoann Pigné, Eric Sanlaville, Jason Schoeters

▶ **To cite this version:**

Stefan Balev, Yoann Pigné, Eric Sanlaville, Jason Schoeters. Temporally connected components. 2023. hal-03966327

HAL Id: hal-03966327

https://hal.science/hal-03966327v1

Preprint submitted on 31 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Temporally connected components

Stefan Balev
Normandie Univ, UNIHAVRE, LITIS, 76600 Le Havre, France

Yoann Pigné
Normandie Univ, UNIHAVRE, LITIS, 76600 Le Havre, France

Eric Sanlaville
Normandie Univ, UNIHAVRE, LITIS, 76600 Le Havre, France

Jason Schoeters
University of Cambridge, United Kingdom

Abstract

We discuss a variety of extensions of connected components in temporal graphs, focusing on extensions using connectivity over time through temporal paths (or journeys). Starting with components induced by temporal sources or sinks, we build up to components induced by multiple sources or sinks, and eventually components where all vertices are sources and sinks, *i.e.* temporally connected components. Our contributions mainly include structural results on the number of components, and algorithmic and complexity results of corresponding decision problems.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Networks → Network components; Mathematics of computing → Graph algorithms; Mathematics of computing → Paths and connectivity problems

Keywords and phrases temporal graph theory, NP-complete problem, polynomial-time reduction, algorithm, connected component, temporal connectivity

Funding *Eric Sanlaville*: DyNet RIN Tremplin Région Normandie 2020-2022
Jason Schoeters: DyNet RIN Tremplin Région Normandie 2020-2022

1 Introduction

Temporal graphs have become increasingly more popular in the literature over the years, and with good reason. Dynamic settings, whether failure-prone systems or highly mobile entities, which static graphs fail to model, can be modeled naturally with temporal graphs [9, 24, 48]. Many problems in temporal graphs can be solved by extending static graph structures and related problems and algorithms into the dimension of time, resulting in various more complicated extensions. The structures considered in this paper use journeys (also called temporal paths) which allow for connectivity over time through increasing time labels on successive edges of the path [12, 25, 28, 33, 35]. Concerning the problems, often these become harder in terms of time and/or space complexity [1, 4, 40]. Naturally, this leads to results considering specific classes of temporal graphs, often restraining the underlying graph (or footprint), approximation results, and/or fixed-parameter tractability results (or, in the negative case, $W[1]$ or even $W[2]$ hardness results) [11, 15, 16, 19, 29, 50]. Temporal graph theory has also natural links with gossip theory [3, 21, 22], and with rainbow structures in edge-colored static graphs [7, 13, 38].

The aim of this paper is to present a comprehensive study of the fundamental question "What is a connected component in temporal graphs?" Some works have appeared over the years offering answers to this question, in one way or another. Here, we wish to regroup these results, not unlike a survey, and add or complete missing pieces of information relevant to this central question. We choose to consider a well-established hierarchy of temporal connectivity properties proposed by Casteigts *et al.* in [9], and recently revisited in [8], as a basis for all sorts of connectivity which may arise or exist in temporal graphs. For some of these properties defined through journeys, we define the corresponding component and present results on computational complexity, algorithms, and the number of components.

In Section 2, we present the main concepts regarding temporal graphs, different types of connectivity such as journeys, and components in such graphs. In Section 3, we state results regarding temporal components in which one vertex connects to/from all others. Then, in Section 4, we present results regarding temporal components in which all vertices connect to/from all others. Finally, in Section 5, we conclude by giving a summary of this paper and discuss some open questions and future works.

Note that as the paper presents a number of new results, for ease of reading, most of the technical proofs are moved to the appendices. The concerned results are marked with a $\star$.

2 Main concepts of temporal graph connectivity

2.1 Temporal graphs

We use standard notation and terminology from graph theory [17]. Temporal graphs can be defined in a number of (more or less) equivalent ways, and as a result exist under a multitude of distinct names, including dynamic graphs, stream graphs, link streams, time-varying graphs and evolving graphs [2, 9, 18, 39]. We choose the following, often referred to as the compact representation.

A temporal graph $\mathcal{G} = (G, \lambda)$ is defined by a static graph $G = (V, E)$ with vertex set $V(G)$ and edge set $E(G) \subseteq V(G) \times V(G)$, and a labelling of the edges $\lambda : E \to 2^{\mathbb{N}} \setminus \emptyset$ determining at which discrete times edges are present. G is commonly called the underlying graph (or footprint) of $\mathcal{G}$, sometimes denoted as $G_{\downarrow}(\mathcal{G})$ or simply $G_{\downarrow}$. In this paper, we consider only finite-time temporal graphs, with the largest label, or lifetime, of $\mathcal{G}$ being denoted as $T(\mathcal{G})$. When there is no ambiguity, we denote $V(G)$, $E(G)$ and $T(\mathcal{G})$ as simply V, E and T respectively, and $n = |V|$ and $m = |E|$. We use the notation $G_t(\mathcal{G})$ (or simply G_t) for the static graph corresponding to temporal graph $\mathcal{G}$ at time t, *i.e.* $G_t = (V, E_t \subseteq E)$ with $e \in E_t \iff t \in \lambda(e)$. G_t is called a snapshot of $\mathcal{G}$. It is sometimes useful to represent temporal graphs as sequences of snapshots, *e.g.* $\mathcal{G} = (G_1, G_2, ..., G_T)$. Let the restriction of $\mathcal{G}$ on time window $[t_1, t_2]$ for some $t_1 \leq t_2 \leq T$ be the temporal graph denoted $\mathcal{G}_{[t_1,t_2]} = (G_{t_1}, G_{t_1+1}, ..., G_{t_2})$, and $G_{[t_1,t_2]}$ denote its footprint. Finally, let the subgraph of $\mathcal{G}$ induced by vertex set $V' \subseteq V$ be the temporal graph denoted $\mathcal{G}[V'] = (G' = (V', E'), \lambda)$, with $\{u, v\} \in E' \iff u, v \in V'$ and $\{u, v\} \in E$.

Size of a temporal graph

Whether one considers the compact representation, or the sequence of snapshots representation, in both cases the size of a temporal graph is $O(n + mT)$. In specific cases, more resourceful representations exist, such as to store only the differences from a snapshot to another (added and removed edges), which is specially effective for temporal graphs with few

changes between snapshots, or low dynamics. In this paper however, we do not specifically consider temporal graphs with low dynamics. Without loss of generality, we assume $m \geq n$.[1] In this setting, an algorithm on temporal graphs is polynomial if and only if it runs in polynomial time regarding n,m, and T.

2.2 Journeys and temporal connectivity

In static graphs, including snapshots of temporal graphs, vertices s, t are adjacent if they are connected by edge $\{s, t\}$, and edges $\{u, v\}, \{v, w\}$ are said to be incident. An (elementary) path is defined as a (non-repeating) sequence of adjacent vertices. We denote a path from u to v as $u \to v$ or, since we consider undirected graphs, $u \leftrightarrow v$. If for all pairs of vertices in G there exists a path including both vertices, then G is said to be connected. If G isn't connected, then G can be partitioned into subgraphs such that each subgraph is connected. Such subgraphs, and equivalently the associated vertex sets, are referred to as the connected components of G.

In temporal graphs, a natural extension of paths exists, in which even though two vertices may admit no path in any of the snapshots, over time they might still be connected thanks to some structures called temporal paths or journeys. More formally, a journey from s to t in $\mathcal{G}$ is a path from s to t in the footprint G with non-decreasing labels on successive edges (see Figure 1). We then say s can reach t and t can be reached by s. We denote a journey, or the existence of a journey, from u to v, as $u \rightsquigarrow v$. Journeys, as opposed to paths, are neither symmetrical nor transitive, *i.e.* $u \rightsquigarrow v$ and $v \rightsquigarrow w$ does not necessarily imply $u \rightsquigarrow w$. The notation $u \rightsquigarrow v \rightsquigarrow w$ implies that journey $v \rightsquigarrow w$ takes place later in time than journey $u \rightsquigarrow v$ (which also implies $u \rightsquigarrow w$). When a temporal graph $\mathcal{G}$ admits journeys from all vertices to all other vertices, the graph is said to be temporally connected, or $\mathcal{TC}$. We use the notation $u \circlearrowleft v$ for a round-trip journey $u \rightsquigarrow v \rightsquigarrow u$

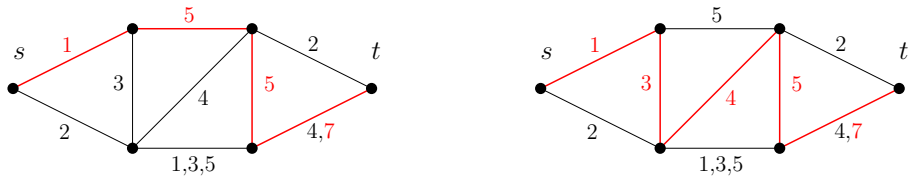


Figure 1 Example of temporal graph $\mathcal{G}_1$ (presented in the compact representation), with two journeys (in red) but no static path from vertex s to vertex t in any of the snapshots.

Besides the footprint, another useful structure concerning temporal graphs is the transitive closure, which represents the reachability (through journeys) of all vertices of the graph. Formally, the transitive closure of a temporal graph $\mathcal{G}$ corresponds to the directed graph $H(\mathcal{G}) = (V, A \subseteq V \times V)$, with $A = \{(v, v') : v \rightsquigarrow v'\}$. Note that this definition introduced by [6] uses the notion of closure in a temporal sense: as stated previously, $u \rightsquigarrow v$ and $v \rightsquigarrow w$ implies $u \rightsquigarrow w$ only if the second journey takes place later than the first one. So $H(\mathcal{G})$ is not, as usual, the transitive closure of some relation, for instance the relation associated to the existence of journeys between vertex couples, but precisely the graph of this relation. When clear from the context, we simply denote $H(\mathcal{G})$ as H.

[1] If for some temporal graph $\mathcal{G}$ we have $m < n$, *i.e.* the footprint admits multiple connected components V', precomputation allows us to reduce to $O(n)$ cases of temporal graphs $\mathcal{G}' = \mathcal{G}[V']$ where $m' \geq n'$.

For example, the transitive closure of the temporal graph $\mathcal{G}_1$ from Figure 1 corresponds to a complete bidirectional graph (all arcs in both directions exist) except for arc (t, s) which is missing, meaning the only journey missing for $\mathcal{G}_1$ to be temporally connected is from t to s.

Constructing the transitive closure can be done in time $O(n(m \log T + n \log n))$ through n calls of an adaptation of Dijkstra's algorithm [46], or in time $O(\texttt{max}(|E_i|)nT)$ through an online algorithm gradually building the transitive closure snapshot by snapshot [5]. The latter is more efficient in specific cases when the lifetime $T = O(n)$ and when the snapshot density $\texttt{max}(|E_i|) = o(m)$. For simplicity, we will consider the former time complexity throughout this paper, and we will refer to the adaptation of Dijkstra's algorithm as the temporal Dijkstra algorithm.

2.3 Hierarchy of connectivity properties for temporal graphs

In [9], and recently revisited in [8], Casteigts *et al.* introduce a hierarchy of temporal properties, including among others temporal connectivity, $\mathcal{TC}$. The properties are given mnemonic names using key concepts *e.g.* $\mathcal{J}$ for "journey", and $\mathcal{TC}$ for "temporal connectivity". Superscript adds restraints, for example $\mathcal{B}$ meaning "in each bounded time window", and $\forall 1$ meaning "from all vertices to one". This gives rise to properties such as $\mathcal{J}^{1\forall}$, having a vertex which is connected through journeys to all other vertices, and $\mathcal{TC}^{\mathcal{B}}$, each window of some given size Δ[2] being temporally connected. The properties of interest for this paper are formally defined in Sections 3 and 4 when we define their corresponding components. For clarity, we've opted to change some mnemonic names of the hierarchy for shorter ones without superscript (so we may add superscript later on). $\mathcal{S}$ for "source" will replace $\mathcal{J}^{1\forall}$, and since "sink" unfortunately also starts with an S, $\mathcal{T}$ for "target" will replace $\mathcal{J}^{\forall 1}$. By slight abuse of notation, we will often use the mnemonic names to denote the property as well as the set of graphs admitting the property.

2.4 From connectivity properties to temporal components

The rest of the paper is organized to define and study, for every journey-based temporal property $\mathcal{X}$, the analogue of a connected component corresponding to this property, which we will simply refer to as a $\mathcal{X}$ component.

▶ **Definition 1** ($\mathcal{X}$ component). *Given a temporal graph $\mathcal{G}$, a $\mathcal{X}$ component is a maximal subset $V' \subseteq V$ such that $\mathcal{X}$ is respected by V' in $\mathcal{G}$.*

Note the maximality requirement for V', which mimics the maximality requirement for static connected components. It is natural to look for largest components also in the temporal case. Theorem 1 can be ambiguous concerning the latter part, since technically $\mathcal{X}$ is a property of temporal graphs, not of a set of vertices in a temporal graph. For now, it is sufficient to say that we simply aim to present the difference between $\mathcal{X}$ components and closed $\mathcal{X}$ components, Theorem 1 and Theorem 2 *resp.* Also, these definitions are formal and clear when presented using concrete properties in Sections 3 and 4.

Using terminology from [6, 23, 44], we say a $\mathcal{X}$ component V' is closed if journeys between vertices $u, v \in V'$ necessary for property $\mathcal{X}$, do not use vertices from $V \setminus V'$. In other words,

[2] In this paper, Δ will by default be used to denote the size of a time window, and $\Delta(G)$ for the maximum degree of a static graph G.

to verify if V' is a closed $\mathcal{X}$ component, it suffices to verify the $\mathcal{X}$ property on $\mathcal{G}[V']$, and this verification does not depend on the rest of $\mathcal{G}$.

▶ **Definition 2** (Closed $\mathcal{X}$ component). *Given a temporal graph $\mathcal{G}$, a closed $\mathcal{X}$ component is a maximal subset $V' \subseteq V$ such that $\mathcal{X}$ is respected by $\mathcal{G}[V']$.*

Open $\mathcal{X}$ components are defined as $\mathcal{X}$ components in which there exists at least one journey necessary for $\mathcal{X}$ which goes outside of the component. Most of the components studied in static graphs are closed (an exception being k-connected components [45] which may be open). We give the following results for $\mathcal{X}$ components, and if applicable, also for closed $\mathcal{X}$ components.

For each property $\mathcal{X}$, we start by studying the worst-case number of $\mathcal{X}$ components, *i.e.* the maximum number of $\mathcal{X}$ components which may exist in a given temporal graph. This can be useful for enumeration and partition problems.

For each property $\mathcal{X}$, the corresponding decision problem $\mathcal{X}$ Component is defined as follows.

▶ **Definition 3** ($\mathcal{X}$ Component decision problem).
Input: *temporal graph $\mathcal{G}$, integer k (and integer Δ if $\mathcal{X}$ is a windowed property).*
Question: *does $\mathcal{G}$ admit a $\mathcal{X}$ component of size at least k?*

Algorithms and complexity results for $\mathcal{X}$ Component are presented. As stated in Section 1, depending on property $\mathcal{X}$, some results may already exist in the literature. However we additionally determine the boundary (if one exists) between polynomial-time solvability and NP-hardness depending on the lifetime of the graph. To obtain such results, we modify reductions from the literature or use different reductions altogether.

Regarding hardness implications, Bhadra *et al.* [6] give the following argument for $\mathcal{TC}$ components and closed $\mathcal{TC}$ components, which we generalize for any temporal properties $\mathcal{X}_1$, $\mathcal{X}_2$. Although $\mathcal{X}_1$ components are a special case of $\mathcal{X}_2$ components, the NP-hardness of $\mathcal{X}_1$ Component does not directly imply that $\mathcal{X}_2$ Component is NP-hard as well. This is because a possible polynomial time algorithm for $\mathcal{X}_2$ Component need only answer the decision problem and not identify the components of size at least k, thus potentially making it difficult to verify if at least one such a component is a $\mathcal{X}_1$ component. Also, the same temporal graph may contain both a $\mathcal{X}_1$ component (of indeterminate size) and a $\mathcal{X}_2$ component of size k, so the decision problem for the latter would always return "yes", ignoring the presence or absence of a $\mathcal{X}_1$ component of size k, thereby leaving its decision problem unsolved. Of course, the other way around, since $\mathcal{X}_1$ components are a special case of $\mathcal{X}_2$ components, if $\mathcal{X}_2$ Component is NP-hard, then $\mathcal{X}_1$ Component is not necessarily NP-hard either. Since hardness results do not transfer one way or the other, note that this implies that neither do results on polynomial-time solvability.

Regarding hardness proofs, *i.e.* reductions, we do however use a trick from [6] making one reduction work for both $\mathcal{X}$ Component and Closed $\mathcal{X}$ Component, which is to make sure all $\mathcal{X}$ components in the transformed instance are closed.

3 One to/from all ($\mathcal{S}$, $\mathcal{S}^{\mathcal{B}}$, $\mathcal{S}^{\mathcal{D}}$, $\mathcal{T}$, $\mathcal{T}^{\mathcal{B}}$, $\mathcal{T}^{\mathcal{D}}$)

A temporal source is a vertex u such that all other vertices v in the temporal graph admit a journey $u \rightsquigarrow v$. The concept is mainly useful to model a network over which one agent,

say the leader, can control the entire network, diffusing information, messages or influence throughout the network using peer-to-peer broadcasting [14, 31, 49]. The property of having a temporal source is denoted by $\mathcal{S}$.

▶ **Definition 4** ($\mathcal{S}$ component)**.** *An $\mathcal{S}$ component of a temporal graph $\mathcal{G}$ is a maximal subset $V' \subseteq V$ such that $\exists u \in V'$, $\forall v \in V'$, $u \rightsquigarrow v$ in $\mathcal{G}$.*

Adding the natural constraint of time windows has mainly two advantages, first a time bound after which one is ensured all other vertices can be reached from the source, and second the possibility for the source to reach other vertices multiple times over the lifetime of the network. The property of having a vertex which is a temporal source for each time window is denoted by $\mathcal{S}^{\mathcal{B}}$.

▶ **Definition 5** ($\mathcal{S}^{\mathcal{B}}$ component)**.** *An $\mathcal{S}^{\mathcal{B}}$ component of duration Δ of a temporal graph $\mathcal{G}$ is a maximal subset $V' \subseteq V$ such that $\exists u \in V'$, $\forall t \leq T - \Delta + 1$, $\forall v \in V'$, $u \rightsquigarrow v$ in $\mathcal{G}_{[t,t+\Delta-1]}$.*

A natural relaxation of the latter allows for any vertex to be the source in a time window, not necessarily the same vertex for all time windows. In other words, the source may be dynamic and change over time. This property is denoted by $\mathcal{S}^{\mathcal{D}}$.

▶ **Definition 6** ($\mathcal{S}^{\mathcal{D}}$ component)**.** *An $\mathcal{S}^{\mathcal{D}}$ component of duration Δ of a temporal graph $\mathcal{G}$ is a maximal subset $V' \subseteq V$ such that $\forall t \leq T - \Delta + 1$, $\exists u \in V'$, $\forall v \in V'$, $u \rightsquigarrow v$ in $\mathcal{G}_{[t,t+\Delta-1]}$.*

Closely related to source vertices are sink vertices, where other vertices are able to reach such a vertex, leading to properties $\mathcal{T}$, $\mathcal{T}^{\mathcal{B}}$, and $\mathcal{T}^{\mathcal{D}}$. While often studied independently and having its own specific applications such as collecting and analysing data (*e.g.* [30, 34, 47]), we simplify this paper through the following observation.

▶ **Lemma 7.** *$\mathcal{T}$ (resp. $\mathcal{S}$, $\mathcal{T}^{\mathcal{B}}$, $\mathcal{S}^{\mathcal{B}}$, $\mathcal{T}^{\mathcal{D}}$, $\mathcal{S}^{\mathcal{D}}$) components in temporal graph $\mathcal{G} = (G, \lambda)$ of lifetime T correspond to $\mathcal{S}$ (resp. $\mathcal{T}$, $\mathcal{S}^{\mathcal{B}}$, $\mathcal{T}^{\mathcal{B}}$, $\mathcal{S}^{\mathcal{D}}$, $\mathcal{T}^{\mathcal{D}}$) components in temporal graph $\mathcal{G}' = (G, \lambda')$ of lifetime T, where $\forall e \in E(G), \lambda'(e) = \bigcup_{\ell \in \lambda(e)} T - \ell$.*

Proof. Any journey in $\mathcal{G}$ from vertex u to v is reversed in $\mathcal{G}'$. Thus, a vertex able to reach all vertices in $\mathcal{G}$ (or in some $\mathcal{G}_{[t,t+\Delta-1]}$) can be reached by all vertices in $\mathcal{G}'$ (or in some $\mathcal{G}'_{[t',t'+\Delta-1]}$) and vice versa. ◀

Theorem 7 allows us to focus only on $\mathcal{S}$ components, since any structural result transfers to $\mathcal{T}$ components, and algorithmic results transfer too, with a polynomial overhead of $O(mT)$ to build the reversed graph (but they can be adapted easily without any overhead). The same holds for $\mathcal{S}^{\mathcal{B}}$ components and $\mathcal{T}^{\mathcal{B}}$ components, as well as for $\mathcal{S}^{\mathcal{D}}$ components and $\mathcal{T}^{\mathcal{D}}$ components.

3.1 $\mathcal{S}$ components

▶ **Lemma 8.** *$\mathcal{S}$ components are necessarily closed.*

Proof. In an $\mathcal{S}$ component, all vertices on a journey from the source to some other vertex also admit a journey from the source, and are thus by maximality included in the component as well. ◀

▶ **Lemma 9.** *A vertex can only be a source for one $\mathcal{S}$ component.*

Proof. If a vertex was a source for two distinct $\mathcal{S}$ components, then at least one wouldn't be maximal since their union would result in a larger $\mathcal{S}$ component. ◀

▶ **Observation 10.** *An isolated vertex in the footprint is an $\mathcal{S}$ component.*

▶ **Theorem 11** ($\star$)**.** *The worst-case number of $\mathcal{S}$ components is n.*

▶ **Theorem 12** ($\star$)**.** *The worst-case number of $\mathcal{S}$ components is $\frac{n}{2}$ in temporal graphs without isolated vertices.*

▶ **Theorem 13** ($\star$)**.** *$\mathcal{S}$ Component is solvable in polynomial time $O(n(m \log T + n \log n))$.*

3.2 $\mathcal{S}^{\mathcal{B}}$ components

▶ **Observation 14.** *Contrary to $\mathcal{S}$ components, $\mathcal{S}^{\mathcal{B}}$ components can be open, as shown in Figure 2.*

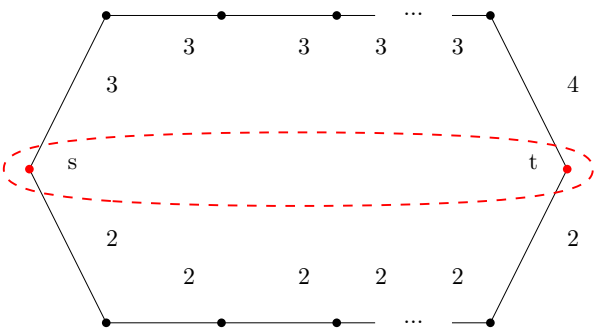


Figure 2 A graph family admitting an open $\mathcal{S}^{\mathcal{B}}$ component for $\Delta = 2$ (in dashed and red with vertex s as source) and an open $\mathcal{T}^{\mathcal{B}}$ component for $\Delta = 2$ (in dashed and red with vertex t as sink).

▶ **Lemma 15.** *A vertex can only be a source for one $\mathcal{S}^{\mathcal{B}}$ component.*

Proof. If a vertex was a source for two distinct $\mathcal{S}^{\mathcal{B}}$ components, then at least one wouldn't be maximal since their union would result in a larger $\mathcal{S}^{\mathcal{B}}$ component. ◀

▶ **Theorem 16** ($\star$)**.** *The worst-case number of $\mathcal{S}^{\mathcal{B}}$ components is n.*

▶ **Theorem 17** ($\star$)**.** *$\mathcal{S}^{\mathcal{B}}$ Component is solvable in polynomial time $O((T - \Delta)n(m \log \Delta + n \log n))$.*

▶ **Theorem 18** ($\star$)**.** *The worst-case number of closed $\mathcal{S}^{\mathcal{B}}$ components is n.*

▶ **Theorem 19** ($\star$)**.** *Closed $\mathcal{S}^{\mathcal{B}}$ Component is solvable in polynomial time $O(n(n-k)(T - \Delta)(m \log \Delta + n \log n))$.*

3.3 $\mathcal{S}^{\mathcal{D}}$ components

Some results from $\mathcal{S}^{\mathcal{B}}$ components transfer for $\mathcal{S}^{\mathcal{D}}$ components, or give bounds, since $\mathcal{S}^{\mathcal{B}}$ components are $\mathcal{S}^{\mathcal{D}}$ components in which the source for every window is the same vertex.

▶ **Observation 20.** *Even $\mathcal{S}^{\mathcal{D}}$ components where sources change between successive windows can be open, such as shown in Figure 3.*

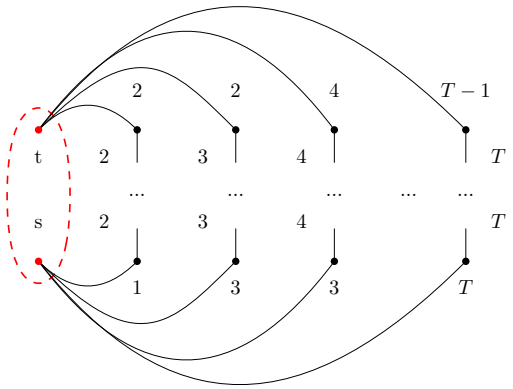


Figure 3 A graph family admitting an open $\mathcal{S}^{\mathcal{D}}$ component for $\Delta = 2$ (in dashed and red with vertex s as source for windows [`odd`, `even`] and t as source for windows [`even`, `odd`]).

▶ **Lemma 21.** *An ordered set of vertices (one source per window possibly with repetitions) corresponds to sources of at most one $\mathcal{S}^{\mathcal{D}}$ component.*

Proof. If such a set of vertices were sources for two distinct $\mathcal{S}^{\mathcal{D}}$ components, then at least one wouldn't be maximal since their union would result in a larger $\mathcal{S}^{\mathcal{D}}$ component. ◀

▶ **Theorem 22** ($\star$)**.** *The worst-case number of $\mathcal{S}^{\mathcal{D}}$ components is at least n, and at most* $\min(2^n, n^{T-\Delta+1})$.

▶ **Theorem 23** ($\star$)**.** *$\mathcal{S}^{\mathcal{D}}$ Component is solvable in time $O(n^{\min(k,T-\Delta+2)}(T-\Delta)k(m\log\Delta + n\log n))$.*

▶ **Theorem 24** ($\star$)**.** *$\mathcal{S}^{\mathcal{D}}$ Component is NP-complete.*

▶ **Theorem 25** ($\star$)**.** *The worst-case number of closed $\mathcal{S}^{\mathcal{D}}$ components is at least n, and at most* $\min(2^n, n^{T-\Delta+1})$.

▶ **Theorem 26** ($\star$)**.** *Closed $\mathcal{S}^{\mathcal{D}}$ Component is solvable in time $O(\min(2^n, n^{T-\Delta+1})(T-\Delta)n(m\log\Delta + n\log n))$.*

▶ **Theorem 27** ($\star$)**.** *Closed $\mathcal{S}^{\mathcal{D}}$ Component is NP-complete.*

4 All to/from all ($\mathcal{TC}$, $\mathcal{TC}^{\mathcal{B}}$, $\mathcal{TC}^{\circlearrowleft}$)

A graph is said to be temporally connected if there exists a journey from each vertex to every other vertex. In other words, over time, each vertex is able to connect to the whole temporal graph, and is used for example in contexts with multi-hop message passing, distributed mobile agents, or social communication networks [26, 36, 37].

▶ **Definition 28** ($\mathcal{TC}$ component)**.** *A $\mathcal{TC}$ component of a temporal graph $\mathcal{G}$ is a maximal subset $V' \subseteq V$ such that $\forall u, v \in V',\ u \rightsquigarrow v$.*

Similar to $\mathcal{S}^{\mathcal{B}}$, time windows add multiple advantages to temporal connectivity. Gomez *et al.* suppose a $\mathcal{TC}^{\mathcal{B}}$ component exists in their work on agreement in dynamic systems in [23]. In [10], Casteigts *et al.* present a general framework for computing parameters in temporal graphs, one of which being Δ for which the given graph is $\mathcal{TC}^{\mathcal{B}}$. Closed $\mathcal{TC}^{\mathcal{B}}$ components

are studied by Huyghues-Despointes *et al.* in [27]. They propose polynomial-time algorithms for computing lower and upper bounds on the maximum component size.

▶ **Definition 29** ($\mathcal{TC}^{\mathcal{B}}$ component). *A $\mathcal{TC}^{\mathcal{B}}$ component of duration Δ of a temporal graph $\mathcal{G}$ is a maximal subset $V' \subseteq V$ such that $\forall v, v' \in V'$, $\forall t \leq T - \Delta + 1$, $v \rightsquigarrow v'$ in temporal graphs $\mathcal{G}_{[t,t+\Delta-1]}$.*

Another parameter Casteigts *et al.* are interested in is the round trip temporal diameter, being the shortest duration for which there exist round trip journeys between every pair of vertices. More generally, round-trip connectivity can represent systems in which feedback or acknowledgements are needed in a connection, such as Transmission Control Protocol (TCP).

▶ **Definition 30** ($\mathcal{TC}^{\circlearrowleft}$ component). *A $\mathcal{TC}^{\circlearrowleft}$ component of a temporal graph $\mathcal{G}$ is a maximal subset $V' \subseteq V$ such that $\forall u, v \in V', u \circlearrowleft v$ in $\mathcal{G}$.*

4.1 $\mathcal{TC}$ components

▶ **Observation 31.** *$\mathcal{TC}$ components can be open such as shown in Figure 4.*

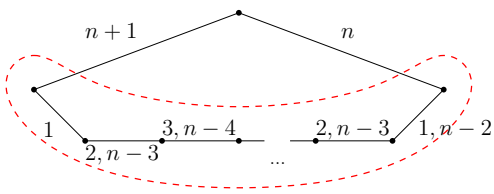


Figure 4 A graph family admitting an open $\mathcal{TC}$ component (in dashed and red).

▶ **Theorem 32** ($\star$). *The worst-case number of $\mathcal{TC}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

Note (see appendix) that the exact bounds are $3^{(\sqrt{n+1}+1)/3}$ and $3^{n/3}$ respectively. We used $2^{0.52k} < 3^{k/3} < 2^{0.53k}$ so that results are comparable with results using powers of 2.

An algorithm is mentioned by Bhadra *et al.* [6] and implemented and experimented on by Nicosia *et al.* [42]. To the best of our knowledge, no complexity analysis of this algorithm has been performed.

▶ **Theorem 33** ($\star$). *$\mathcal{TC}$ Component is solvable in time $O(nm \log T + \min(n^k k^2, 2^{0.25n}))$.*

Bhadra *et al.* prove that $\mathcal{TC}$ Component is NP-complete. They use a reduction from Clique in which they produce a temporal graph with a lifetime $T = 4$ (so using labels between 1 and 4 included). This leaves the NP-hardness question open for lifetimes $1 < T < 4$ ($T = 1$ being polynomial-time solvable). We treat the case of $T = 2$, filling the gap and proving general NP-hardness for $T > 1$.

▶ **Theorem 34** ($\star$). *$\mathcal{TC}$ Component is NP-complete, for all constant lifetimes $T > 1$.*

▶ **Theorem 35** ($\star$). *The worst-case number of closed $\mathcal{TC}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

▶ **Theorem 36** ($\star$). *Closed $\mathcal{TC}$ Component is solvable in time $O(2^n n(m \log T + n \log n))$.*

Bhadra *et al.* also prove Closed $\mathcal{TC}$ Component is NP-complete through the same reduction as for $\mathcal{TC}$ Component. We use the same trick to show Closed $\mathcal{TC}$ Component is NP-complete for all constant lifetimes $T > 1$.

▶ **Theorem 37** ($\star$). *Closed $\mathcal{TC}$ Component is NP-complete, for all constant lifetimes $T > 1$.*

4.2 $\mathcal{TC}^{\mathcal{B}}$ components

▶ **Lemma 38.** *$\mathcal{TC}^{\mathcal{B}}$ components can be open.*

Proof. Since $\mathcal{TC}$ components are $\mathcal{TC}^{\mathcal{B}}$ components for the specific setting of $\Delta = T$, Theorem 32 transfers directly. ◀

▶ **Theorem 39** ($\star$). *The worst-case number of $\mathcal{TC}^{\mathcal{B}}$ components is at least $2^{0.52\sqrt{n}}$ and at most $2^{0.53n}$.*

▶ **Theorem 40** ($\star$). *$\mathcal{TC}^{\mathcal{B}}$ Component is solvable in time $O((T-\Delta)n(m\log\Delta + n\log n) + \min(n^k k^2, 2^{0.25n}))$.*

Again, since $\mathcal{TC}$ components are $\mathcal{TC}^{\mathcal{B}}$ components by setting window size $\Delta = T$, NP-hardness transfers. We however prove a more general hardness for $\mathcal{TC}^{\mathcal{B}}$ Component regarding any constant lifetime and window size.

▶ **Theorem 41** ($\star$). *$\mathcal{TC}^{\mathcal{B}}$ Component is NP-complete, for all constant lifetimes $T > 1$ and window sizes $\Delta > 1$.*

▶ **Theorem 42** ($\star$). *The worst-case number of closed $\mathcal{TC}^{\mathcal{B}}$ components is at least $2^{0.52\sqrt{n}}$ and at most $2^{0.53n}$.*

▶ **Theorem 43** ($\star$). *Closed $\mathcal{TC}^{\mathcal{B}}$ Component is solvable in time $O(2^n(T-\Delta)n(m\log\Delta + n\log n))$.*

▶ **Theorem 44** ($\star$). *Closed $\mathcal{TC}^{\mathcal{B}}$ Component is NP-complete, for all constant lifetimes $T > 1$ and window sizes $\Delta > 1$.*

4.3 $\mathcal{TC}^{\circlearrowleft}$ components

▶ **Observation 45.** *$\mathcal{TC}^{\circlearrowleft}$ components can be open, such as shown in Figure 5.*

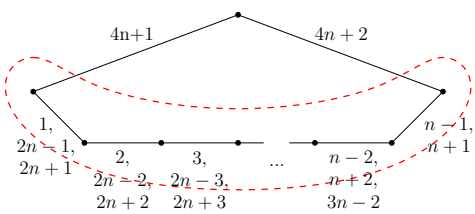


Figure 5 Graph family admitting an open $\mathcal{TC}^{\circlearrowleft}$ component (in red and dashed).

▶ **Theorem 46** ($\star$). *The worst-case number of $\mathcal{TC}^{\circlearrowleft}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

▶ **Theorem 47** ($\star$). *$\mathcal{TC}^{\circlearrowleft}$ Component is solvable in time $O(n^2(m\log T + n\log n) + \min(n^k k^2, 2^{0.25n}))$.*

Outside of the trivial case of lifetime $T = 1$, $\mathcal{TC}^{\circlearrowleft}$ Component can be solved in polynomial time for lifetime $T = 2$ as well.

▶ **Theorem 48** ($\star$). *$\mathcal{TC}^{\circlearrowleft}$ Component is solvable in polynomial time $O(n+m)$ on temporal graphs with constant lifetimes $T \leq 2$.*

▶ **Theorem 49** ($\star$). *$\mathcal{TC}^{\circlearrowleft}$ Component is NP-complete, for all constant lifetimes $T > 2$.*

▶ **Theorem 50** ($\star$). *The worst-case number of closed $\mathcal{TC}^{\circlearrowleft}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

▶ **Theorem 51** ($\star$). *Closed $\mathcal{TC}^{\circlearrowleft}$ Component is solvable in time $O(2^n(n^2(m \log T + n \log n))$.*

▶ **Theorem 52** ($\star$). *Closed $\mathcal{TC}^{\circlearrowleft}$ Component is solvable in polynomial time $O(n+m)$ on temporal graphs with constant lifetimes $T \leq 2$.*

▶ **Theorem 53** ($\star$). *Closed $\mathcal{TC}^{\circlearrowleft}$ Component is NP-complete, for all constant lifetimes $T > 2$.*

5 Conclusion

5.1 Summary of this paper

A first summary of this paper is given in the following table, where we precise for each $\mathcal{X}$ component the worst-case number of components, or the presented bounds on this number (using notation [`lower bound`, `upper bound`]), as well as the complexity of the corresponding decision problem and its corresponding complexity class (P referring to polynomial-time solvable, NPC to NP-complete). For $\mathcal{X}$ components defined through the $\mathcal{TC}$ property (lower part of the table) we give the constant values of T (and Δ when applicable) for which the corresponding decision problem is NP-complete.

Component	Worst-case Number	Complexity	Class
$\mathcal{S}$	n ($\frac{n}{2}$ no isolated vertices)	$O(n(m \log T + n \log n))$	P
$\mathcal{S}^{\mathcal{B}}$	n	$O((T-\Delta)n(m \log \Delta + n \log n))$	P
Closed $\mathcal{S}^{\mathcal{B}}$	n	$O(n(T-\Delta)(n-k)(m \log \Delta + n \log n))$	P
$\mathcal{S}^{\mathcal{D}}$	$[n, \min(2^n, n^{T-\Delta+1})]$	$O(n^{\min(k,T-\Delta+2)}(T-\Delta)k(m \log \Delta + n \log n))$	NPC
Closed $\mathcal{S}^{\mathcal{D}}$	$[n, \min(2^n, n^{T-\Delta+1})]$	$O(\min(2^n, n^{T-\Delta+1})(T-\Delta)n(m \log \Delta + n \log n))$	NPC
$\mathcal{TC}$	$[2^{0.52\sqrt{n}}, 2^{0.53n}]$	$O(nm \log T + \min(n^k k^2, 2^{0.25n}))$	NPC ($T > 1$)
Closed $\mathcal{TC}$	$[2^{0.52\sqrt{n}}, 2^{0.53n}]$	$O(2^n n(m \log T + n \log n))$	NPC ($T > 1$)
$\mathcal{TC}^{\mathcal{B}}$	$[2^{0.52\sqrt{n}}, 2^{0.53n}]$	$O((T-\Delta)n(m \log \Delta + n \log n) + \min(n^k k^2, 2^{0.25n}))$	NPC ($T\&\Delta > 1$)
Closed $\mathcal{TC}^{\mathcal{B}}$	$[2^{0.52\sqrt{n}}, 2^{0.53n}]$	$O(2^n(T-\Delta)n(m \log \Delta + n \log n))$	NPC ($T\&\Delta > 1$)
$\mathcal{TC}^{\circlearrowleft}$	$[2^{0.52\sqrt{n}}, 2^{0.53n}]$	$O(n^2(m \log T + n \log n) + \min(n^k k^2, 2^{0.25n}))$	NPC ($T > 2$)
Closed $\mathcal{TC}^{\circlearrowleft}$	$[2^{0.52\sqrt{n}}, 2^{0.53n}]$	$O(2^n(n^2(m \log T + n \log n))$	NPC ($T > 2$)

Temporal graphs with lifetime $T = 1$ trivially admit polynomial-time algorithms for all these problems (since they reduce to finding connected components in the static graph), so they are not presented. Note that all these results are to be found in this paper, although some parts were first presented in related works. We also prove $\mathcal{TC}^{\circlearrowleft}$ Component and Closed $\mathcal{TC}^{\circlearrowleft}$ Component can be solved in polynomial time for temporal graphs with lifetime $T = 2$.

5.2 A short discussion about optimisation of windowed components

This paper provided algorithms (polynomial or not) to solve the windowed versions of our problems. Some of these consider treat the windows in the specific order $([1, \Delta], [2, \Delta + 1], ...[T - \Delta + 1, T])$. We note this order isn't necessary, in the sense that the result would remain unchanged if it considered the time windows in other orders. A consequence potentially useful for optimisation is that one could reorder the time windows in the most "favorable" way. For example, for $\mathcal{S}^{\mathcal{B}}$, start with the window in which the corresponding temporal graph may have few and small candidate components, and start the intersection process with windows in which the corresponding temporal graphs may have very distinct $\mathcal{S}$ components from the candidate components. Of course, theoretically this would only be useful if such an estimation and reordering of time windows could be done in time $O((T - \Delta)(n(m \log \Delta + n \log n)))$ as well, and even then practically this may still induce a significant change in running time.

Another possible optimisation concerns some bounded versions of problems related to $\mathcal{TC}$, where transitive closures are computed for each time window. Can one do better by computing a modified transitive closure over the whole graph, which iteratively keeps track of how old journeys' starting dates are, and removes them over time if too old? Our first investigations did not allow to show this idea brings a breakthrough.

Both of these directions we leave as open research avenues in this paper.

5.3 A quick note on parameterised complexity

Among the NP-hard problems in this paper, all but two are para-NP-hard concerning the lifetime parameter, *i.e.* for some constant value of T the problem is NP-hard. The two problems which are not para-NP-hard regarding lifetime T are $\mathcal{S}^{\mathcal{D}}$ Component and Closed $\mathcal{S}^{\mathcal{D}}$ Component. Both are in XP, *i.e.* the problems are polynomial-time solvable for all constant T (and $\mathcal{S}^{\mathcal{D}}$ Component is in XP as well concerning the solution size parameter k). It may be of interest to determine where exactly in XP the problems are situated, *e.g.* FPT or W[1]-hard. Of course, outside of T, other natural parameters may be considered. Again, this research is left as open avenues.

References

1. Eleni C Akrida, Leszek Gasieniec, George B Mertzios, and Paul G Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61(3):907–944, 2017.
2. Aris Anagnostopoulos, Ravi Kumar, Mohammad Mahdian, Eli Upfal, and Fabio Vandin. Algorithms on evolving graphs. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 149–160, 2012.
3. Brenda Baker and Robert Shostak. Gossips and telephones. *Discrete Mathematics*, 2(3):191–193, June 1972.
4. Stefan Balev, Yoann Pigné, Eric Sanlaville, and Mathilde Vernet. Complexité du problème de Steiner dynamique. In *23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, Villeurbanne - Lyon, France, February 2022. INSA Lyon.
5. Matthieu Barjon, Arnaud Casteigts, Serge Chaumette, Colette Johnen, and Yessin M Neggaz. Testing temporal connectivity in sparse dynamic graphs. *arXiv preprint arXiv:1404.7634*, 2014.
6. Sandeep Bhadra and Afonso Ferreira. Computing multicast trees in dynamic networks and the complexity of connected components in evolving graphs. *Journal of Internet Services and Applications*, 3(3):269–275, 2012.
7. Peter Bradshaw and Bojan Mohar. A Rainbow Connectivity Threshold for Random Graph Families. In Jaroslav Nešetřil, Guillem Perarnau, Juanjo Rué, and Oriol Serra, editors, *Extended Abstracts EuroComb 2021*, Trends in Mathematics, pages 842–847. Springer International Publishing, 2021.
8. Arnaud Casteigts. *A Journey Through Dynamic Networks (with Excursions)*. Habilitation à diriger des recherches, Université de Bordeaux, June 2018.
9. Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.
10. Arnaud Casteigts, Ralf Klasing, Yessin M Neggaz, and Joseph G Peters. Computing parameters of sequence-based dynamic graphs. *Theory of Computing Systems*, 63(3):394–417, 2019.
11. Arnaud Casteigts, Joseph G Peters, and Jason Schoeters. Temporal cliques admit sparse spanners. *Journal of Computer and System Sciences*, 121:1–17, 2021.
12. Arnaud Casteigts, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp thresholds in random simple temporal graphs. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–326. IEEE, 2022.
13. Lily Chen, Xueliang Li, and Yongtang Shi. The complexity of determining the rainbow vertex-connection of a graph. *Theoretical Computer Science*, 412(35):4531–4535, 2011.
14. Julia Chuzhoy and Sanjeev Khanna. A new algorithm for decremental single-source shortest paths with applications to vertex-capacitated flow and cut problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 389–400. Association for Computing Machinery, 2019.
15. Pierluigi Crescenzi, Clémence Magnien, and Andrea Marino. Approximating the temporal neighbourhood function of large temporal graphs. *Algorithms*, 12(10):211, 2019.
16. Argyrios Deligkas, Eduard Eiben, and George Skretas. Minimizing reachability times on temporal graphs via shifting labels. *arXiv preprint arXiv:2112.08797*, 2021.

17 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Berlin, Heidelberg, 2017.

18 Antoine Dutot, Frédéric Guinand, Damien Olivier, and Yoann Pigné. Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. In *Emergent Properties in Natural and Artificial Complex Systems. (ECCS'2007)*, 2007.

19 Jessica Enright, Kitty Meeks, and Hendrik Molter. Counting temporal paths. *arXiv preprint arXiv:2202.12055*, 2022.

20 Jeff Erickson. *Algorithms*. self-publishing, 2019.

21 Satoshi Fujita, Stephane Perennes, and Joseph G Peters. Neighbourhood gossiping in hypercubes. *Parallel Processing Letters*, 8(02):189–195, 1998.

22 Frits Göbel, J Orestes Cerdeira, and Hendrik Jan Veldman. Label-connected graphs and the gossip problem. *Discrete Mathematics*, 87(1):29–40, 1991.

23 Carlos Gómez-Calzado, Arnaud Casteigts, Alberto Lafuente, and Mikel Larrea. A connectivity model for agreement in dynamic systems. In Jesper Larsson Träff, Sascha Hunold, and Francesco Versaci, editors, *Euro-Par 2015: Parallel Processing*, Lecture Notes in Computer Science, pages 333–345, Berlin, Heidelberg, 2015. Springer, Springer.

24 John P. Hayes. A graph model for fault-tolerant computing systems. *IEEE Transactions on Computers*, 25(09):875–884, 1976.

25 Anne-Sophie Himmel, Hendrik Molter, Rolf Niedermeier, and Manuel Sorge. Adapting the bron–kerbosch algorithm for enumerating maximal cliques in temporal graphs. *Social Network Analysis and Mining*, 7(1):35, 2017.

26 Petter Holme. Network reachability of real-world contact sequences. *Physical Review E*, 71(4):046119, 2005.

27 Charles Huyghues-Despointes, Binh-Minh Bui-Xuan, and Clémence Magnien. Forte delta-connexité dans les flots de liens. In *ALGOTEL 2016-18èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, 2016.

28 Allen Ibiapina and Ana Silva. Mengerian temporal graphs revisited. In *International Symposium on Fundamentals of Computation Theory*, pages 301–313. Springer, Springer International Publishing, 2021.

29 David Ilcinkas, Ralf Klasing, and Ahmed Mouhamadou Wade. Exploration of constantly connected dynamic graphs based on cactuses. In *International Colloquium on Structural Information and Communication Complexity*, pages 250–262. Springer, 2014.

30 Joe-Air Jiang, Chien-Hao Wang, Min-Sheng Liao, Xiang-Yao Zheng, Jen-Hao Liu, Cheng-Long Chuang, Che-Lun Hung, and Chia-Pang Chen. A wireless sensor network-based monitoring system with dynamic convergecast tree algorithm for precision cultivation management in orchid greenhouses. *Precision agriculture*, 17(6):766–785, 2016.

31 Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks. *arXiv preprint arXiv:2007.03113*, 2020.

32 Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, Boston, MA, 1972.

33 David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.

34 Abdul Waheed Khan, Abdul Hanan Abdullah, Mohammad Hossein Anisi, and Javed Iqbal Bangash. A comprehensive study of data collection schemes using mobile sinks in wireless sensor networks. *Sensors*, 14(2):2510–2548, 2014.

35 Nina Klobas, George B Mertzios, Hendrik Molter, and Paul G Spirakis. The complexity of computing optimum labelings for temporal connectivity. *arXiv preprint arXiv:2202.05880*, 2022.

36 Gueorgi Kossinets, Jon Kleinberg, and Duncan Watts. The structure of information pathways in a social communication network. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 435–443. Association for Computing Machinery, August 2008.

37 Vassilis Kostakos. Temporal graphs. *Physica A: Statistical Mechanics and its Applications*, 388(6):1007–1023, 2009.

38 Michael Krivelevich and Raphael Yuster. The rainbow connection of a graph is (at most) reciprocal to its minimum degree. *Journal of Graph Theory*, 63(3):185–191, 2010.

39 Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):61, 2018.

40 George B Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. Computing maximum matchings in temporal graphs. *arXiv preprint arXiv:1905.05304*, 2019.

41 John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965.

42 Vincenzo Nicosia, John Tang, Mirco Musolesi, Giovanni Russo, Cecilia Mascolo, and Vito Latora. Components in time-varying graphs. *Chaos: An interdisciplinary journal of nonlinear science*, 22(2):023101, 2012.

43 John M Robson. Finding a maximum independent set in time o (2n/4). Technical report, Technical Report 1251-01, LaBRI, Université Bordeaux I, 2001.

44 Mathilde Vernet, Yoann Pigné, and Éric Sanlaville. A study of connectivity on dynamic graphs: Computing persistent connected components. *4OR*, April 2022.

45 Dong Wen, Lu Qin, Ying Zhang, Lijun Chang, and Ling Chen. Enumerating k-Vertex Connected Components in Large Graphs. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 52–63, April 2019.

46 B Bui Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02):267–285, 2003.

47 Hiroyuki Yomo, Akitoshi Asada, and Masato Miyatake. On-demand data gathering with a drone-based mobile sink in wireless sensor networks exploiting wake-up receivers. *IEICE Transactions on Communications*, 101(10):2094–2103, 2018.

48 Ping Yu, Zhiping Wang, Peiwen Wang, Haofei Yin, and Jia Wang. Dynamic evolution of shipping network based on hypergraph. *Physica A: Statistical Mechanics and its Applications*, 598:127247, July 2022.

49 Wenyu Zang, Peng Zhang, Chuan Zhou, and Li Guo. Discovering Multiple Diffusion Source Nodes in Social Networks. *Procedia Computer Science*, 29:443–452, 2014.

50 Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. The complexity of finding small separators in temporal graphs. *Journal of Computer and System Sciences*, 107:72–92, 2020.

A Auxiliary proofs

A.1 $\mathcal{S}$ components

▶ **Theorem 11.** *The worst-case number of $\mathcal{S}$ components is n.*

Proof. Theorem 9 implies that at most n $\mathcal{S}$ components can exist, which by Theorem 10 is tight when considering the trivial case of the empty temporal graph, *i.e.* a temporal graph composed of empty graph snapshots $G_i = (V, \emptyset)$. ◀

For Theorem 12, we establish a relation between $\mathcal{S}$ components and locally minimum label edge sets, *i.e.* connected edge sets sharing a label which have no incident edges with lower labels other than themselves.

▶ **Lemma 54.** *The number of $\mathcal{S}$ components is at most the number of locally minimum edge sets in a temporal graph without isolated vertices.*

Proof. For any locally minimum label edge set $E' = \{\{u_1, u_2\}, ...\{u_j, u_k\}\}$, observe that $\mathcal{S}$ components S_{u_i} with source $u_i \in V(E')$ are identical (in terms of their vertex set), since all vertices u_i can reach all vertices reachable by all other vertices of E' through the locally minimum label edge set. Hence, such vertices u_i of a locally minimum label edge set produce altogether at most one $\mathcal{S}$ component. Concerning vertices which are not part of a locally minimum label edge set, say vertex v, observe that since v is not part of a locally minimum label edge set, this means that its incident edge with the smallest label, say $\{v, x\}$, must have some incident edge $\{x, y\}$ with an even smaller label (or $\{v, x\}$ is part of a connected edge set of the same label which has such an incident edge). Now, either $\{x, y\}$ is part of a locally minimum label edge set or it has again some incident edge with a smaller label (or again its connected edge set of the same label does). Continue this process until having found a locally minimum label edge set, say $E' = \{\{u_1, u_2\}, ...\{u_j, u_k\}\}$. Note that any u_i can reach v at the latter's earliest incident edge's label, implying any vertex v can reach, can be reached by u_i as well, through v. $\mathcal{S}$ component S_v with source v thus has to be contained (in terms of their vertex set) within the larger $\mathcal{S}$ component S_{u_i} with source u_i, since u_i can reach all vertices v can reach. Hence, vertices v which are not part of a locally minimum label edge set cannot be the source of a $\mathcal{S}$ component, since it would not be maximal. This combined with locally minimum label edge sets producing at most one $\mathcal{S}$ component each, effectively bounds the number of $\mathcal{S}$ components by the number of locally minimum label edge sets. ◀

Note that there is not necessarily equality for Theorem 54, for example when one component induced by a locally minimum label edge set is included in another, such as shown in Figure 6.

▶ **Lemma 55.** *At most $\frac{n}{2}$ locally minimum label edge sets may exist in any temporal graph, which is tight.*

Proof. Since locally minimum label edge sets cannot be incident by definition, and cover at least two vertices each, at most $\frac{n}{2}$ locally minimum label edge sets exist in any graph. This is shown tight as follows. Take a complete graph on n vertices $K_n = (V, E)$ without any labels. Now, take a maximum matching in K_n and assign small labels to it, and to the rest of the edges, either assign large labels or remove these edges. The result is a temporal graph with exactly $\frac{n}{2}$ locally minimum label edge sets, corresponding to the maximum matching's $\frac{n}{2}$ edges.

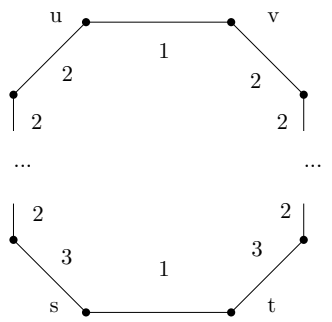


Figure 6 Example graph family for which Theorem 54 does not induce an equality between the number of $\mathcal{S}$ components (only one exists composed of all vertices with source either u or v) and the number of locally minimum edge sets (two such sets exist: $\{u, v\}$ and $\{s, t\}$).

◀

▶ **Theorem 12.** *The worst-case number of $\mathcal{S}$ components is $\frac{n}{2}$ in temporal graphs without isolated vertices.*

Proof. In other words, we prove that at most $\frac{n}{2}$ $\mathcal{S}$ components may exist in any temporal graph without isolated vertices, and we prove this is tight. Figure 7 shows a graph family which admits $\frac{n}{2}$ $\mathcal{S}$ components, which by Theorem 54 and Theorem 55 finishes the proof.

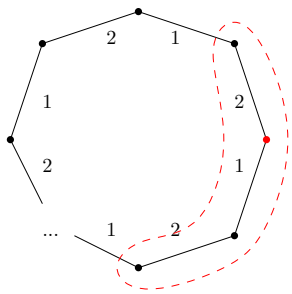


Figure 7 A graph family with $\frac{n}{2}$ $\mathcal{S}$ components. A component with its corresponding source is marked in red.

◀

▶ **Theorem 13.** *$\mathcal{S}$ Component is solvable in polynomial time $O(n(m \log T + n \log n))$.*

Proof. We propose the following algorithm for $\mathcal{S}$ Component. First, construct the transitive closure H of the given temporal graph $\mathcal{G}$, then check if $k \leq \Delta^+(H) + 1$, where $\Delta^+(H)$ denotes the maximum outdegree of H. The described algorithm has a time complexity of $O(n(m \log T + n \log n))$ to construct the transitive closure H, and an additional $O(n^2)$ to compute the maximum outdegree $\Delta^+(H)$. ◀

▶ **Theorem 56.** *For a given temporal graph, all $\mathcal{S}$ components can be enumerated in time $O(nm \log T + n^3)$.*

Proof. Start by computing the transitive closure H, taking time $O(n(m \log T + n \log n))$. Then, let the set of $\mathcal{S}$ components be composed of vertex v with all its out-neighbours in H, for all vertices $v \in V$. Now, some of these $\mathcal{S}$ components may be included in others, meaning they may be non-maximal. To remove non-maximal $\mathcal{S}$ components, start by sorting the vertices in each $\mathcal{S}$ component, taking time $O(n^2 \log n)$. Then, for each pair of $\mathcal{S}$ components, of which there are $O(n^2)$, check whether one is a subset of the other, and remove it if so, which can be done in time $O(n)$. All in all, enumeration of $\mathcal{S}$ components can be done in time $O(n(m \log T + n \log n) + n^2 \log n + n^3) = O(nm \log T + n^3)$. ◀

A.2 $\mathcal{S}^{\mathcal{B}}$ components

▶ **Theorem 16.** *The worst-case number of $\mathcal{S}^{\mathcal{B}}$ components is n.*

Proof. Theorem 15 implies at most n $\mathcal{S}^{\mathcal{B}}$ components may exist, which is tight for example for the empty temporal graph. In fact, it is tight even for much denser graphs; a temporal graph with one window of size Δ containing only empty snapshots is sufficient to obtain n $\mathcal{S}^{\mathcal{B}}$ components of size 1, meaning even a temporal graph with a complete graph as footprint may attain n $\mathcal{S}^{\mathcal{B}}$ components. ◀

▶ **Theorem 17.** *$\mathcal{S}^{\mathcal{B}}$ Component is solvable in polynomial time $O((T - \Delta)n(m \log \Delta + n \log n))$.*

Proof. We propose the following algorithm for $\mathcal{S}^{\mathcal{B}}$ Component. Start by enumerating the $\mathcal{S}$ components of the windowed temporal graph $\mathcal{G}_{[1,\Delta]}$, which by Theorem 56 can be done in time $O(nm \log \Delta + n^3)$. However, we are not interested in maximal $\mathcal{S}$ components, so the added $O(n^3)$ can be removed. Thus, we enumerate $\mathcal{S}$ components, some of which may not be maximal, in time $O(n(m \log \Delta + n \log n))$. Keep only those of size at least k. These are candidate components for $\mathcal{S}^{\mathcal{B}}$ components. Let us denote these candidate components as S_s by their unique corresponding source vertex s (Theorem 9). Then, for each windowed temporal graph $\mathcal{G}_{[t,t+\Delta-1]}$, of which there are $O(T-\Delta)$, obtain their $\mathcal{S}$ components in the same manner, again in time $O(n(m \log \Delta + n \log n))$. Denote these $\mathcal{S}$ components as $S_s[t, t + \Delta - 1]$. For each candidate component S_s, of which there are $O(n)$, update S_s by setting it to the intersection of S_s and $S_s[t, t + \Delta - 1]$ which is doable in time $O(n \log n)$. Remove S_s from the candidate components if its size is less than k. If no candidate components remain at some point, the instance is negative, otherwise it is positive. All in all, this gives a total time complexity of $O(n(m \log \Delta + n \log n) + (T - \Delta)n(m \log \Delta + n \log n + n(n \log n))) = O((T - \Delta)n(m \log \Delta + n \log n))$. ◀

▶ **Theorem 57.** *For a given temporal graph, all $\mathcal{S}^{\mathcal{B}}$ components can be enumerated in time $O((T - \Delta)n(m \log T + n \log n) + n^3)$.*

Proof. Run the algorithm from Theorem 17 and at the end add the verification and removal of non-maximal $\mathcal{S}^{\mathcal{B}}$ components, doable as in Theorem 56 in time $O(n^3)$, for a total of $O((T - \Delta)n(m \log T + n \log n) + n^3)$. ◀

▶ **Theorem 18.** *The worst-case number of closed $\mathcal{S}^{\mathcal{B}}$ components is n.*

Proof. Theorem 16 directly adapts for closed $\mathcal{S}^{\mathcal{B}}$ components, since n $\mathcal{S}^{\mathcal{B}}$ components of size 1 are necessarily closed. ◀

▶ **Lemma 58.** *Verifying a solution vertex subset S_s of source s and of size k for* Closed $\mathcal{S}^{\mathcal{B}}$ Component *takes time $O((T-\Delta)(\min(m,k^2)\log\Delta + k\log k))$.*

Proof. For each windowed temporal subgraph $\mathcal{G}[S_s]_{[t,t+\Delta-1]}$, of which there are $O(T-\Delta)$, apply the temporal Dijkstra algorithm from vertex s to obtain its reachability which is done in time $O(\min(m,k^2)\log\Delta + k\log k)$. If for all windows, its reachability covers the whole vertex set S_s, then S_s is a solution, else it is not. ◀

▶ **Theorem 19.** *Closed $\mathcal{S}^{\mathcal{B}}$ Component is solvable in polynomial time $O(n(n-k)(T-\Delta)(m\log\Delta + n\log n))$.*

Proof. We propose the following algorithm for Closed $\mathcal{S}^{\mathcal{B}}$ Component. Run the algorithm for $\mathcal{S}^{\mathcal{B}}$ Component. At the end of the algorithm, the set of candidates contains all $\mathcal{S}^{\mathcal{B}}$ components S_s of size at least k. For all S_s, test if it is closed, *i.e.* if for all $t \leq T-\Delta+1$, S_s is an $\mathcal{S}$ component of temporal graph $\mathcal{G}[S_s]_{[t,t+\Delta-1]}$. If some component is, then the instance is positive. If however S_s is open, meaning at some time $t \leq T-\Delta+1$, vertex s cannot reach all vertices in $\mathcal{G}[S_s]_{[t,t+\Delta-1]}$, then S_s can be shrunk down to contain only the reachable vertices in $\mathcal{G}[S_s]_{[t,t+\Delta-1]}$. This shrunken down S_s can now be tested again: if it is of size at least k and closed, then the instance is positive, and if not we can again shrink S_s *etc.* If no closed $\mathcal{S}^{\mathcal{B}}$ component remains of suitable size, then the instance is negative. The algorithm for $\mathcal{S}^{\mathcal{B}}$ Component runs in time $O((T-\Delta)n(m\log\Delta+n\log n))$, after which $O(n)$ candidate components are to test whether they contain a large enough closed $\mathcal{S}^{\mathcal{B}}$ component. For each of these, testing if they are closed, takes time $O((T-\Delta)(m\log\Delta+n\log n))$ by Theorem 58. This process is repeated in case the component is not closed and a shrunken component has to be tested which again isn't closed but contains another shrunken component to be tested *etc.* This can happen at most $n-k$ times, since the size of a component is at most n and we're not interested in components of size less than k. Hence, all together, we obtain a complexity of $O(n(T-\Delta)(m\log\Delta+n\log n) + n(n-k)(T-\Delta)(m\log\Delta+n\log n)) = O(n(n-k)(T-\Delta)(m\log\Delta+n\log n))$. ◀

A.3 $\mathcal{S}^{\mathcal{D}}$ components

▶ **Theorem 22.** *The worst-case number of $\mathcal{S}^{\mathcal{D}}$ components is at least n, and at most $\min(2^n, n^{T-\Delta+1})$.*

Proof. The lower bound comes from Theorem 18, with an example being the empty temporal graph. For the upper bounds, since $\mathcal{S}^{\mathcal{D}}$ components are subsets of vertices which may intersect, at most 2^n $\mathcal{S}^{\mathcal{D}}$ components may exist. By Theorem 21, representing components through their sources, *i.e.* ordered sets of vertices (one source per window), another bound of $n^{T-\Delta+1}$ is found. ◀

▶ **Lemma 59.** *Verifying a solution vertex subset V' of size k for $\mathcal{S}^{\mathcal{D}}$ Component takes time $O((T-\Delta)k(m\log\Delta+n\log n))$.*

Proof. For each window $[t, t+\Delta-1]$, of which there are $T-\Delta$, one constructs a partial transitive closure H' of $\mathcal{G}_{[t,t+\Delta-1]}$ in which only reachability for the vertices V' is computed. This takes time $O(k(m\log\Delta+n\log n))$. For each induced subgraph $H'[V']$, checking if a vertex with outdegree $k-1$ exists takes time $O(k^2)$. In total, this takes time $O((T-\Delta)(k(m\log\Delta+n\log n)+k^2) = O((T-\Delta)k(m\log\Delta+n\log n))$. ◀

▶ **Theorem 23.** *$\mathcal{S}^{\mathcal{D}}$ Component is solvable in time $O(n^{\min(k,T-\Delta+2)}(T-\Delta)k(m\log\Delta + n\log n))$.*

Proof. We present two algorithms, the first being the brute force algorithm of testing all possible subsets of vertices of size k, the other a generalization of the algorithm for $\mathcal{S}^{\mathcal{B}}$ Component. Let us start by analysing the brute force algorithm first. For all possible subsets of vertices of interest, of which there exist $\binom{n}{k} = O(n^k)$, we verify if it is a suitable solution. By Theorem 59, one verification takes time $O((T-\Delta)k(m\log\Delta+n\log n))$, meaning the brute force algorithm runs in time $O(n^k(T-\Delta)k(m\log\Delta + n\log n))$. Concerning the generalization of the algorithm for $\mathcal{S}^{\mathcal{B}}$ Component, only one change needs to be made. Instead of shrinking candidate component S_s at every window by taking the union with $S_s[t, t+\Delta-1]$, we will need to instead shrink the candidate component with all $S_v[t, t+\Delta-1]$ for all $v \in S_s$ (which could all correspond to a source for that time window), splitting a candidate component up into potentially $O(n)$ distinct components every time window. A result of this is that at every time window $[t, t+\Delta-1]$, not $O(n)$ candidate components S_s exist from previous windows, but at most $n^t = O(n^{T-\Delta+1})$ candidate components $S_{(s,t,u,\ldots,v)}$. Analysing the complexity gives $O(n^{T-\Delta+1}n(T-\Delta)(m\log\Delta + n\log n))$. ◀

▶ **Theorem 24.** *$\mathcal{S}^{\mathcal{D}}$ Component is NP-complete.*

Proof. Theorem 59 proves $\mathcal{S}^{\mathcal{D}}$ Component is in NP. To prove NP-hardness, we give a polynomial-time reduction from Clique to $\mathcal{S}^{\mathcal{D}}$ Component, *i.e.* we show how to transform any instance of Clique to an instance of $\mathcal{S}^{\mathcal{D}}$ Component such that the instance of Clique is positive if and only if the instance of $\mathcal{S}^{\mathcal{D}}$ Component is positive.

Given an instance of Clique, being a graph G_1 of n_1 vertices and m_1 edges and an integer k_1, we describe the following polynomial-time transformation to an instance of $\mathcal{S}^{\mathcal{D}}$ Component, being a temporal graph $\mathcal{G}_2 = (G_2, \lambda_2)$ of lifetime T_2 and integers Δ_2 and k_2. Set $T_2 = 3n_1 - 1$, $\Delta_2 = 2$ and $k_2 = m_1 + n_1 + k_1$. Let's construct the footprint G_2, initially the empty graph. Suppose $V(G_1) = \{v_1, v_2, \ldots, v_{n_1}\}$. For each vertex $v_i \in V(G_1)$, add vertices v_i and $v_{i,i}$ with edge $\{v_i, v_{i,i}\}$ to G_2. We refer to vertices v_i in G_2 as original vertices, and to vertices $v_{i,i}$ as satellite vertices. Then, for each edge $\{v_i, v_j\} \in E(G_1)$, add vertex $v_{i,j} = v_{j,i}$ and edges $\{v_i, v_{i,j}\}$, $\{v_{i,j}, v_j\}$ to G_2. These edges will allow for v_i to reach v_j and/or vice versa, depending on the labelling. We refer to vertices $v_{i,j}$ as intermediary vertices. Finally, $\forall v_{i,j}, v_{h,k} \in G_2$, add edges $\{v_{i,j}, v_{h,k}\}$ (essentially creating a clique of all intermediary and satellite vertices, of size $m_1 + n_1$). Let us refer to these last edges as the background edges. This concludes the construction of G_2. Note that $n_2 = m_1 + 2n_1$. Concerning λ_2 (see also Figure 8), for each vertex $v_a \in V(G_1)$ with $1 \le a \le n_1$, we create a temporal graph $\mathcal{G}_{v_a}$ using labels $3a-2$ and $3a-1$. Afterwards, we take the union of these temporal graphs and add some extra labels to all edges to obtain $\mathcal{G}_2$. For now, construct each $\mathcal{G}_{v_a}$ by placing label $3a-1$ on all edges $\{v_i, v_{i,a}\}$ and on the background edges, and label $3a-2$ on all other edges. Note that the intermediary vertices allow for v_i and v_j to reach each other, except for temporal graphs $\mathcal{G}_{v_i}$ in which only v_i can reach v_j, and vice versa for $\mathcal{G}_{v_j}$. Now, let $\mathcal{G}_2$ be the union of all $\mathcal{G}_{v_a}$, for all $1 \le a \le n_1$, and add labels $3a$ on all edges, for all $1 \le a \le n_1 - 1$. This concludes the polynomial-time transformation.

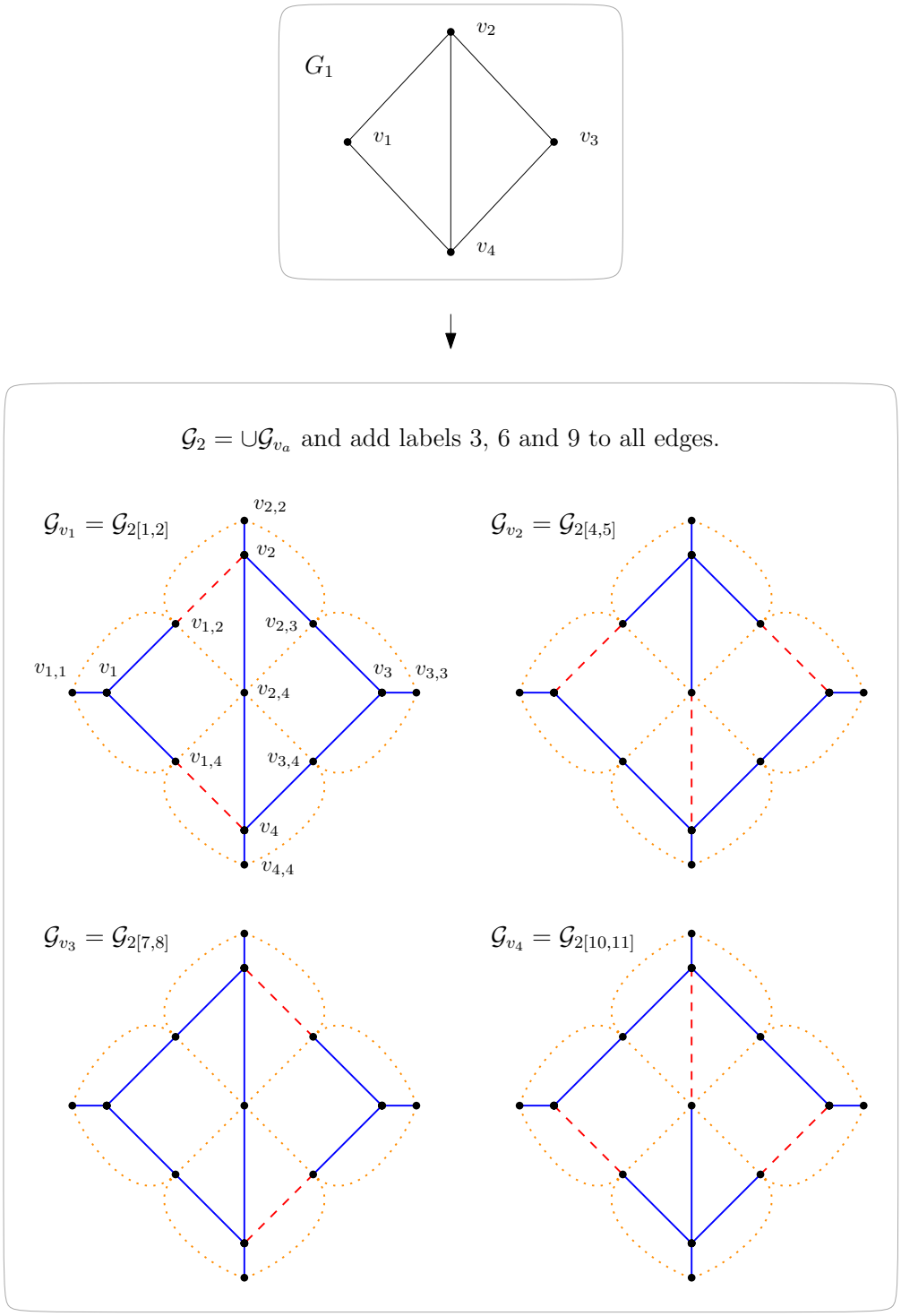


Figure 8 Example of transformation from a CLIQUE instance graph G_1 to a $\mathcal{S}^{\mathcal{D}}$ COMPONENT instance temporal graph $\mathcal{G}_2$. For visibility, for all $\mathcal{G}_{v_a}$, blue full edges correspond to edges with label $3a-2$, red dashed edges to edges with label $3a-1$, and orange dotted edges to background edges with label $3a-1$. Also for visibility, not all background edges are represented.

Now let's show that the instance of CLIQUE, being (G_1, k_1), is positive if and only if the instance of $\mathcal{S}^{\mathcal{D}}$ COMPONENT, being $(\mathcal{G}_2, \Delta_2, k_2)$, is positive.

(G_1, k_1) is positive $\implies$ $(\mathcal{G}_2, \Delta_2, k_2)$ is positive:
Suppose that a clique exists in G_1 of size k_1, composed of vertices $V' = v_h$, $v_{h+1}, ...v_{h+k_1}$. Note that in all windows of size $\Delta_2 = 2$ in $\mathcal{G}_2$, either the window contains $3a$ for some a and thus all vertices can reach each other in this window using only edges with label $3a$, or the window does not contain $3a$ in which case it must contain $3a - 1$, meaning $m_1 + n_1$ vertices can reach each other by using the background edges with label $3a - 1$. In either case, this implies a $\mathcal{S}^{\mathcal{D}}$ component of size at least $m_1 + n_1$ must exist in $\mathcal{G}_2$. To prove a larger $\mathcal{S}^{\mathcal{D}}$ component exists, of size $k_2 = m_1 + n_1 + k_1$, we can ignore windows containing $3a$, since in these windows a $\mathcal{S}$ component of size $n_2 \geq k_2$ exists and can thus be shrunk to adapt to any $\mathcal{S}^{\mathcal{D}}$ component suitable for the other windows. The remaining windows correspond exactly to the temporal graphs $\mathcal{G}_{v_i}$ for all $1 \leq i \leq n_1$. Now, all edges $\{v_i, v_j\}$ composing our clique in G_1 exist by definition as edges $\{v_i, v_{i,j}\}$ and $\{v_{i,j}, v_j\}$, with either labels 1 and 1 *resp.* or 1 and 2 *resp.*, or 2 and 1 *resp.* in temporal graphs $\mathcal{G}_{v_a}$. Let us iterate over these temporal graphs (with variable b), in which two cases are distinguished:

- If the edges of the clique in G_1 are all transformed in edges with label 1 in the temporal graph $\mathcal{G}_{v_b}$, then vertex v_h is a source able to reach V' in $\mathcal{G}_{v_b}$, for a total of at least $m_1 + n_1 + k_1 = k_2$ vertices.
- If some edges of the clique in G_1 are transformed into edges with label 2 in the temporal graph $\mathcal{G}_{v_b}$, then v_b must be a part of the clique in G_1, since by definition only adjacent edges to v_b are transformed into edges with label 2 in $\mathcal{G}_{v_b}$ (outside of background edges). Also, v_b (and only v_b amongst the original vertices) corresponds to a source able to reach V' in $\mathcal{G}_{v_b}$, for a total of at least $m_1 + n_1 + k_1 = k_2$ vertices.

In both cases, and thus for all $\mathcal{G}_{v_a}$, a $\mathcal{S}$ component exists containing the aforementioned $m_1 + n_1$ vertices, as well as the k_1 vertices from the clique in G_1. The result of a $\mathcal{S}^{\mathcal{D}}$ component of size $m_1 + n_1 + k_1 = k_2$ existing in $\mathcal{G}_2$ follows directly.

$(\mathcal{G}_2, \Delta_2, k_2)$ is positive $\implies$ (G_1, k_1) is positive:
Again, observe that any $\mathcal{S}^{\mathcal{D}}$ component in $\mathcal{G}_2$ must contain the $m_1 + n_1$ vertices which the background edges cover. Let us look at the other $k_2 - (m_1 + n_1) = k_1$ vertices. These vertices must be v_i for some i (since vertices $v_{i,j}$ are already contained through the background edges), meaning they correspond directly to vertices v_i of G_1. Let's denote these vertices as V'. Consider temporal graph $\mathcal{G}_{v_b}$ for some v_b, and observe that any vertex v_i can, by construction, only reach other vertices v_j by traversing intermediary vertices $v_{i,j}$. Also, at most one original vertex v_j can be reached by a journey starting at a vertex v_i. This implies all vertices in V' must be linked pairwise through intermediary vertices in $\mathcal{G}_{v_b}$. By construction, intermediary vertices between vertices V' can only exist in $\mathcal{G}_{v_b}$ if the corresponding edges exist between V' in G_1. Thus a clique of size $|V'| = k_1$ must exist in G_1. ◀

▶ **Theorem 25.** *The worst-case number of closed $\mathcal{S}^{\mathcal{D}}$ components is at least n, and at most* $\min(2^n, n^{T-\Delta+1})$.

Proof. The lower bound from Theorem 22 works as $\mathcal{S}^{\mathcal{D}}$ components of size 1 are closed. The upper bounds transfer directly from Theorem 22. ◀

▶ **Lemma 60.** *Verifying a solution vertex subset V' of size k for* Closed $\mathcal{S}^{\mathcal{D}}$ Component *takes time $O((T-\Delta)k(\min(m,k^2)\log\Delta + k\log k))$.*

Proof. For each window $[t, t+\Delta-1]$, of which there are $T-\Delta$, constructing the transitive closure of $\mathcal{G}[V']_{[t,t+\Delta-1]}$ takes time $O(k(\min(m,k^2)\log\Delta + k\log k))$. In each transitive closure, checking if a vertex with outdegree $k-1$ exists takes time $O(k^2)$. In total, this takes time $O((T-\Delta)(k(\min(m,k^2)\log\Delta + k\log k) + k^2) = O(k(T-\Delta)(\min(m,k^2)\log\Delta + k\log k))$. ◀

▶ **Lemma 61.** *The existence of a closed $\mathcal{S}^{\mathcal{D}}$ component of size $> k$ does not necessarily imply the existence of a (non-maximal) closed $\mathcal{S}^{\mathcal{D}}$ component of size k.*

Proof. Consider Figure 9, in which an infinite family of temporal graphs is presented. For each of these graphs, say $\mathcal{G}$, a closed $\mathcal{S}^{\mathcal{D}}$ component exists of size n, with any vertex as source for both windows. However, no (non-maximal) closed $\mathcal{S}^{\mathcal{D}}$ component exists of size $n-1$, since considering any such set of vertices V' would result in either $G(\mathcal{G}[V']_{[1,\Delta]})$ or $G(\mathcal{G}[V']_{[2,\Delta+1]})$ to be disconnected, implying no temporal source can exist in that window.

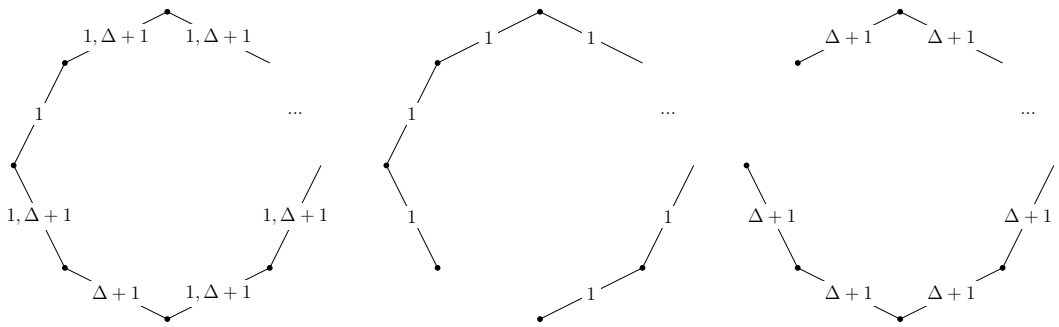


Figure 9 Temporal graph family in which a closed $\mathcal{S}^{\mathcal{D}}$ component of size n exists but no non-maximal component of size $n-1$ exists. The temporal graphs corresponding to time window $[1,\Delta]$ and time window $[2,\Delta+1]$ are shown in the middle and on the right respectively.

◀

▶ **Theorem 26.** *Closed $\mathcal{S}^{\mathcal{D}}$ Component is solvable in time $O(\min(2^n, n^{T-\Delta+1})(T-\Delta)n(m\log\Delta + n\log n))$.*

Proof. Due to Theorem 61, a brute force algorithm cannot simply verify all possible subsets of size k, as a closed $\mathcal{S}^{\mathcal{D}}$ component of size $> k$ may still exist even if one of size k does not. Hence, all possible subsets of size at least k will have to be checked, of which there are $O(\min(2^n, n^{T-\Delta+1}))$ by Theorem 25. Verifying a solution takes time $O((T-\Delta)k(\min(m,k^2)\log\Delta + k\log k))$ by Theorem 60, meaning the brute force algorithm runs in time $O(\min(2^n, n^{T-\Delta+1})(T-\Delta)n(m\log\Delta + n\log n))$. ◀

▶ **Theorem 27.** *Closed $\mathcal{S}^{\mathcal{D}}$ Component is NP-complete.*

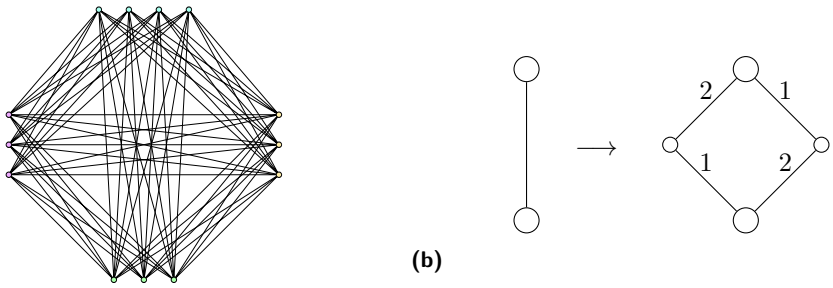
Proof. To prove Closed $\mathcal{S}^{\mathcal{D}}$ Component is in NP, we show how to verify a solution $V' \subseteq V$ in polynomial time. For every window $[t, t+\Delta-1]$, compute the transitive closure of temporal graph $\mathcal{G}[V']_{[t,t+\Delta-1]}$ and check if at least one source, *i.e.* a vertex able to reach all others, exists. If this is the case in all windows, then V' is a solution, and else it is not.

To prove NP-hardness, the same reduction as in Theorem 24 is used, since all the $\mathcal{S}^{\mathcal{D}}$ components in the transformed instance of the reduction are closed $\mathcal{S}^{\mathcal{D}}$ components. ◀

A.4 $\mathcal{TC}$ components

▶ **Theorem 32.** *The worst-case number of $\mathcal{TC}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

Proof. The lower bound has been obtained by Casteigts *et al.* [8] by adapting a Moon and Moser graph [41] on n vertices and $\binom{n}{2} - n$ edges admitting $3^{n/3}$ cliques. Every edge of this graph is replaced by a semaphore gadget (see Figure 10) creating a temporal graph on $N = n^2 - 2n$ vertices in which every initial clique is now a $\mathcal{TC}$ component, obtaining a total of $3^{n/3} = 3^{(\sqrt{N+1}+1)/3} > 3^{\sqrt{N}/3} > 2^{0.52\sqrt{N}}$ $\mathcal{TC}$ components.



(a) **(b)**

Figure 10 A graph with at least $2^{0.52\sqrt{n}}$ $\mathcal{TC}$ components is constructed by taking a Moon and Moser graph (left) admitting $3^{n/3}$ cliques, and replacing each edge with a semaphore gadget (right). All initial cliques now correspond to $\mathcal{TC}$ components, which are maximal when including semaphore vertices.

Outside of the trivial upper bound of 2^n, a better upper bound can be obtained as follows. Since $\mathcal{TC}$ components of $\mathcal{G}$ correspond to bidirectional cliques in the transitive closure of $\mathcal{G}$ (see [6]), modify the transitive closure as follows. For all arcs (u, v) such that (v, u) exists as well, replace both arcs by one (undirected) edge $\{u, v\}$. All other arcs are removed. The corresponding graph is undirected and cliques in this graph correspond to $\mathcal{TC}$ components. By Moon and Moser [41], at most $3^{n/3}$ cliques can exist in undirected graphs, which implies the same upper bound $3^{n/3} < 2^{0.53n}$ holds for $\mathcal{TC}$ components in temporal graphs. ◀

▶ **Theorem 33.** *$\mathcal{TC}$ Component is solvable in time $O(nm \log T + \min(n^k k^2, 2^{0.25n}))$.*

Proof. The algorithm starts by constructing the transitive closure of the input temporal graph. This is done in time $O(n(m \log T + n \log n))$. Then, it searches for bidirectional cliques (a subset of vertices with arcs in both directions between vertices) in the transitive closure. This has the same asymptotic complexity as searching for cliques. Since we're interested in cliques of size at least k, the brute force algorithm of testing each subset of size k runs in time $O(n^k k^2)$. Also, one can obtain a maximum-size clique in time $O(2^{n/4})$ [43]. All together, $\mathcal{TC}$ Component can thus be solved in time $O(n(m \log T + n \log n) + \min(n^k k^2, 2^{n/4})) = O(nm \log T + \min(n^k k^2, 2^{0.25n}))$. ◀

▶ **Theorem 34.** *$\mathcal{TC}$ Component is NP-complete, for all constant lifetimes $T > 1$.*

Proof. We use a specific version of Clique, in which one has to decide if a clique of size $k = \frac{n}{3}$ exists in a given graph on n vertices. This specific problem was indirectly proven NP-hard by Erickson [20] through his reduction from 3-SAT to IndependentSet (in [32], Karp reduces SAT to Clique in a similar fashion). Let us refer to this problem as $\frac{n}{3}$Clique.

We show how to reduce $\frac{n}{3}$ Clique to $\mathcal{TC}$ Component in polynomial time. Given a graph $G = (V, E)$ for the $\frac{n}{3}$ Clique problem, create temporal graph $\mathcal{G}$ with (initially) the same vertex set V. For each edge $(u, v) \in E$, add a semaphore gadget between u and v in $\mathcal{G}$, with semaphore "intermediary" vertices uv and vu (see Figure 11). See Figure 12 for an example of how to construct $\mathcal{G}$.

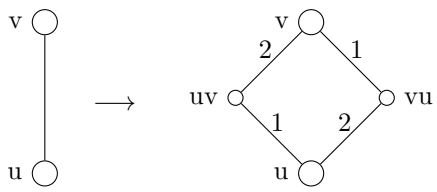


Figure 11 An edge is replaced by a semaphore gadget, adding two intermediary vertices.

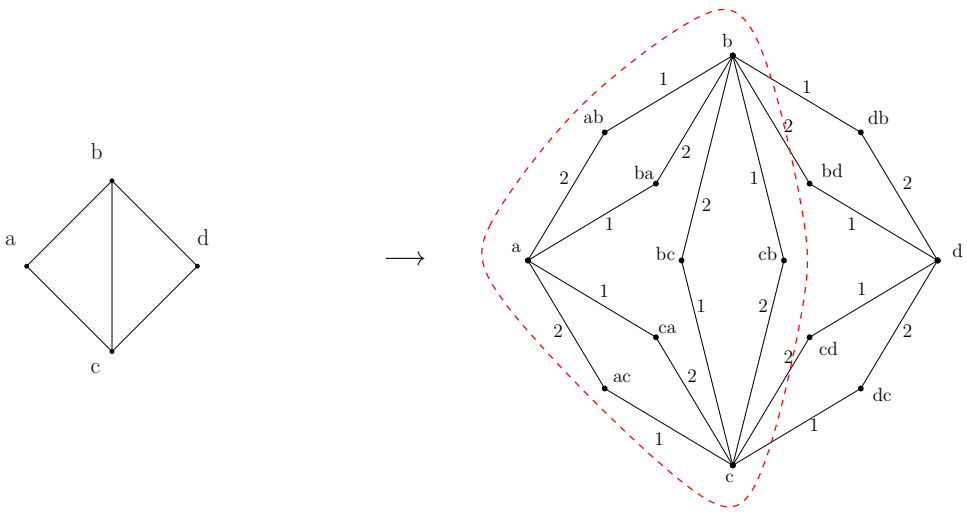


Figure 12 On the left an example instance of Clique with a clique of size 3 which, when transformed as shown on the right, allows for a $\mathcal{TC}$ component of size 9 (shown in red and dashed).

We prove a clique of size $k = \frac{n}{3}$ exists in G if and only if a $\mathcal{TC}$ component of size $k^2 = \frac{n^2}{9}$ exists in $\mathcal{G}$.

$\frac{n}{3}$ Clique $\implies$ $\mathcal{TC}$ Component

If a clique of size k exists in G of vertices $K = \{v_1, v_2, ..., v_k\}$, then consider the corresponding vertices in $\mathcal{G}$, as well as intermediary vertices $\{v_i v_j : v_i, v_j \in K, v_i \neq v_j\}$. Let's refer to this set of vertices in $\mathcal{G}$ as V', which is of size $k + 2\frac{k(k-1)}{2} = k^2$. V' is a $\mathcal{TC}$ component, since all vertices can reach each other:

- a vertex v_i can reach all other vertices v_j through journey $(v_i, v_i v_j, v_j)$ using time steps $(1, 2)$ respectively;
- a vertex v_i can reach all intermediary vertices $v_j v_k$, through journey $(v_i, v_i v_k, v_k, v_j v_k)$ with time steps $(1, 2, 2)$ respectively;

- an intermediary vertex v_iv_j can reach all vertices v_k through journey $(v_iv_j, v_i, v_iv_k, v_k)$ using time steps $(1, 1, 2)$ respectively;
- an intermediary vertex v_iv_j can reach all other intermediary vertices v_kv_ℓ through journey $(v_iv_j, v_i, v_iv_\ell, v_\ell, v_kv_\ell)$ using time steps $(1, 1, 2, 2)$ respectively.

$\mathcal{TC}$ Component $\implies$ $\frac{n}{3}$ Clique

If a $\mathcal{TC}$ component V' of size $k^2 = \frac{n^2}{9}$ exists in $\mathcal{G}$, then consider all vertices $K = V' \cap V$. Trivially $|K| > 0$, furthermore $|K| > 1$ since otherwise the lone vertex $v \in K$ is part of a $\mathcal{TC}$ component of size at most $2\Delta + 1$ (including all adjacent intermediary vertices), which leads to the asymptotic contradiction $k^2 = \frac{n^2}{9} \leq 2\Delta + 1 < 2n + 1$. We prove that K forms a clique of size k in G:

- for all pairs of vertices $v_i, v_j \in K$, intermediary vertex v_iv_j (*resp.* v_jv_i) must be included in V', since otherwise no journey $v_i \rightsquigarrow v_j$ (*resp.* $v_j \rightsquigarrow v_i$) exists in V';
- for all $v_k \notin K$, vertices v_kv_ℓ (*resp.* $v_\ell v_k$) cannot reach (*resp.* be reached by) vertices v_iv_j where $v_i, v_j \in V'$, thus $v_kv_\ell \notin V'$ (*resp.* $v_\ell v_k \notin V'$);
- since V' is composed solely of vertices K and all their corresponding intermediary vertices, we have that $|V'| = k^2 = |K| + 2\frac{|K|(|K|-1)}{2} = |K|^2$, leading to $|K| = k$;
- finally, since all intermediary vertices of K are present in V' which can only be created through a semaphore gadget, which in turn can only be created if a corresponding edge is present in G, K forms a clique in G.

◀

▶ **Theorem 35.** *The worst-case number of closed $\mathcal{TC}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

Proof. Theorem 32 directly adapts for closed components, as the components obtained by the lower bound's construction are closed. ◀

▶ **Lemma 62.** *Verifying a solution vertex subset V' of size k for* Closed $\mathcal{TC}$ Component *takes time $O(k(\min(m, k^2) \log T + k \log k))$.*

Proof. To verify a solution V', we construct the transitive closure of temporal graph $\mathcal{G}[V']$ and check if it is a (bidirectional) complete graph, doable in time $O(k(\min(m, k^2) \log T + k \log k) + k^2) = O(k(\min(m, k^2) \log T + k \log k))$. ◀

▶ **Lemma 63.** *The existence of a closed $\mathcal{TC}$ component of size $> k$ does not necessarily imply the existence of a (non-maximal) closed $\mathcal{TC}$ component of size k.*

Proof. Consider Figure 13, in which an infinite family of temporal graphs is presented. For each of these graphs, say $\mathcal{G}$, a closed $\mathcal{TC}$ component exists of size n. However, no (non-maximal) closed $\mathcal{TC}$ component exists of size $n - 1$, since considering any such set of vertices $V \setminus u_i$ (*resp.* v_i) would result in v_i (*resp.* u_i) not being able to reach any vertex v_j (*resp.* u_j).

◀

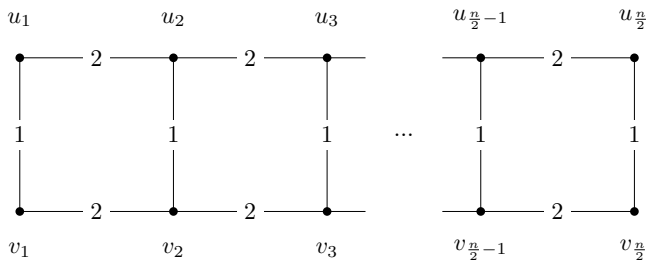


Figure 13 Temporal graph family in which a closed $\mathcal{TC}$ component of size n exists but no non-maximal component of size $n-1$ exists.

▶ **Theorem 36.** *Closed $\mathcal{TC}$ Component is solvable in time $O(2^n n(m \log T + n \log n))$.*

Proof. We describe the brute force algorithm of testing all possible subsets of vertices. Due to Theorem 63, we cannot only test subsets of size k, needing to possibly test all subsets of size at least k, of which there are $O(2^n)$. Testing a vertex subset takes time $O(n(m \log T + n \log n))$ by Theorem 62, giving a total time complexity of $O(2^n n(m \log T + n \log n))$.

◀

▶ **Theorem 37.** *Closed $\mathcal{TC}$ Component is NP-complete, for all constant lifetimes $T > 1$.*

Proof. By Theorem 62, a solution vertex subset can be verified in time $O(n(m \log T + n \log n))$, and thus Closed $\mathcal{TC}$ Component is in NP.

In the construction from Theorem 34, all $\mathcal{TC}$ components in $\mathcal{G}$ are closed. This suffices to prove that Open $\mathcal{TC}$ Component is NP-hard as well, for all lifetimes $T > 1$. ◀

A.5 $\mathcal{TC}^{\mathcal{B}}$ components

▶ **Theorem 39.** *The worst-case number of $\mathcal{TC}^{\mathcal{B}}$ components is at least $2^{0.52\sqrt{n}}$ and at most $2^{0.53n}$.*

Proof. The lower bound transfers directly from Theorem 32.

The upper bound is obtained in a similar manner as for Theorem 32. Start by creating an adaptation of the transitive closure, in which an arc (u, v) is present if and only if a journey exists from u to v in every window of size Δ. (This can be computed easily using temporal Dijkstra algorithms from [46] in each windowed graph.) Then, for all arcs (u, v) such that (v, u) exists as well, replace both arcs by one (undirected) edge $\{u, v\}$. All other arcs are removed. The corresponding graph is undirected and cliques in this graph correspond to $\mathcal{TC}^{\mathcal{B}}$ components. By Moon and Moser [41], at most $3^{n/3}$ cliques can exist in undirected graphs, which implies the same upper bound $3^{n/3} < 2^{0.53n}$ holds for $\mathcal{TC}^{\mathcal{B}}$ components in temporal graphs. ◀

▶ **Theorem 40.** *$\mathcal{TC}^{\mathcal{B}}$ Component is solvable in time $O((T - \Delta)n(m \log \Delta + n \log n) + \min(n^k k^2, 2^{0.25n}))$.*

Proof. For all windowed temporal graphs $\mathcal{G}_{[t,t+\Delta-1]}$, construct transitive closures $H_{[t,t+\Delta-1]}$, taking time $O((T-\Delta)(n(m\log\Delta + n\log n)))$. Afterwards, take the intersection of these transitive closures, denoted by H. An arc (u,v) in H means that u can reach v in all windowed temporal graphs. A bidirectional clique of size k in H thus represents a $\mathcal{TC}^{\mathcal{B}}$ component of size k. As described in Theorem 33, such a clique of size at least k can be found in time $O(\min(n^k k^2, 2^{n/4}))$. All in all, $\mathcal{TC}^{\mathcal{B}}$ Component can thus be solved in time $O((T-\Delta)(n(m\log\Delta + n\log n)) + \min(n^k k^2, 2^{0.25n}))$. ◀

▶ **Theorem 41.** *$\mathcal{TC}^{\mathcal{B}}$ Component is NP-complete, for all constant lifetimes $T > 1$ and window sizes $\Delta > 1$.*

Proof. We prove that instances of $\mathcal{TC}$ Component are equivalent to firstly, instances of $\mathcal{TC}^{\mathcal{B}}$ Component such that Δ is arbitrarily larger than T, and secondly, instances of $\mathcal{TC}^{\mathcal{B}}$ Component such that Δ is arbitrarily smaller than T. We also prove these cover the precised bounds of $T > 1$ and $\Delta > 1$.

Concerning $\Delta \geq T$, we trivially have that any instance $(\mathcal{G}, k)$ for $\mathcal{TC}$ Component, where the lifetime of $\mathcal{G}$ is T, is equivalent to instance $(\mathcal{G}, k, \Delta \geq T)$ for $\mathcal{TC}^{\mathcal{B}}$ Component. This combined with Theorem 34 proves that $\mathcal{TC}^{\mathcal{B}}$ Component is NP-hard for all $T > 1$ and all $\Delta > 1$ such that $\Delta \geq T$.

Concerning $\Delta < T$, any instance $(\mathcal{G}, k)$ for $\mathcal{TC}$ Component, where the lifetime of $\mathcal{G}$ is T, is equivalent to instance $(\mathcal{G}', k, \Delta = T)$ for $\mathcal{TC}^{\mathcal{B}}$ Component, where $\mathcal{G}'$ corresponds to $\mathcal{G}$ with some additional complete snapshots, *i.e.* $\mathcal{G}' = \mathcal{G}^{\frown}(G_i = K_n)_{i=T+1}^{j}$ for some arbitrary $j > T$. Again with Theorem 34, this proves that $\mathcal{TC}^{\mathcal{B}}$ Component is NP-hard for all $T > 1$ and all $\Delta > 1$ such that $\Delta < T$. ◀

▶ **Theorem 42.** *The worst-case number of closed $\mathcal{TC}^{\mathcal{B}}$ components is at least $2^{0.52\sqrt{n}}$ and at most $2^{0.53n}$.*

Proof. The lower bound from Theorem 32 transfers since the construction produces closed $\mathcal{TC}$ components, which are also closed $\mathcal{TC}^{\mathcal{B}}$ components for $\Delta = T$. The upper bounds from Theorem 39 transfer directly. ◀

▶ **Lemma 64.** *Verifying if a vertex subset V' of size k is a solution for Closed $\mathcal{TC}^{\mathcal{B}}$ Component takes time $O((T-\Delta)k(\min(m, k^2)\log\Delta + k\log k))$.*

Proof. For each window $[t, t+\Delta-1]$, compute the transitive closure $H_{[t,t+\Delta-1]}$ of temporal graph $\mathcal{G}[V']_{[t,t+\Delta-1]}$. V' is a solution for Closed $\mathcal{TC}^{\mathcal{B}}$ Component if and only if all $H_{[t,t+\Delta-1]}$ are bidirectional complete graphs. ◀

▶ **Theorem 43.** *Closed $\mathcal{TC}^{\mathcal{B}}$ Component is solvable in time $O(2^n(T-\Delta)n(m\log\Delta + n\log n))$.*

Proof. We present the brute force algorithm. Due to Theorem 63, we know it may not be sufficient to verify only the subsets of size exactly k; we may need to verify all subsets of size at least k, of which there exist $O(2^n)$. By Theorem 64, verifying such a subset requires $O((T-\Delta)n(m\log\Delta + n\log n))$ time. In total, the brute force algorithm thus takes time $O(2^n(T-\Delta)n(m\log\Delta + n\log n))$. ◀

▶ **Theorem 44.** *Closed $\mathcal{TC}^{\mathcal{B}}$ Component is NP-complete, for all constant lifetimes $T > 1$ and window sizes $\Delta > 1$.*

Proof. By Theorem 64, a vertex subset can be verified to be a solution for Closed $\mathcal{TC}^{\mathcal{B}}$ Component in time $O((T-\Delta)n(m \log \Delta + n \log n))$. Closed $\mathcal{TC}^{\mathcal{B}}$ Component is thus in NP.

In the reduction used in Theorem 41, all $\mathcal{TC}^{\mathcal{B}}$ components in the transformed instance are closed. This suffices to prove that Closed $\mathcal{TC}^{\mathcal{B}}$ Component is NP-hard, for any constant lifetime $T > 1$ and window size $\Delta > 1$. ◀

A.6 $\mathcal{TC}^{\circlearrowleft}$ components

▶ **Theorem 46.** *The worst-case number of $\mathcal{TC}^{\circlearrowleft}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

Proof. The lower bound can be obtained as follows. Take a Moon and Moser graph [41] on n vertices and $\binom{n}{2} - n$ edges admitting $3^{n/3}$ cliques. Every edge of this graph is replaced by a semaphore gadget (see Figure 10) creating a temporal graph on $N = n^2 - 2n$ vertices in which every initial clique is now a $\mathcal{TC}^{\circlearrowleft}$ component, obtaining a total of $3^{n/3} > 2^{0.52\sqrt{N}}$ $\mathcal{TC}^{\circlearrowleft}$ components.

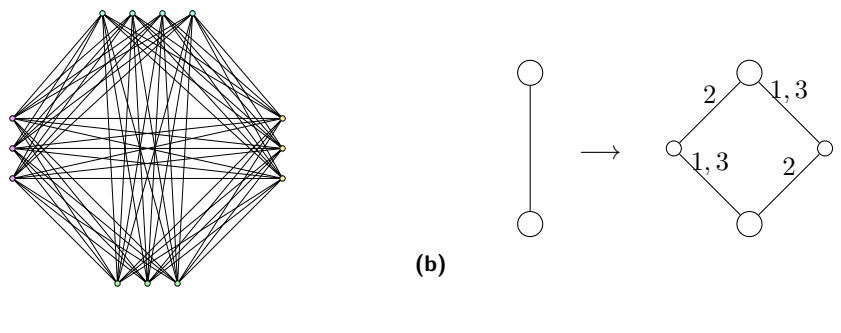


Figure 14 A graph with at least $2^{0.52\sqrt{n}}$ $\mathcal{TC}^{\circlearrowleft}$ components is constructed by taking a Moon and Moser graph (left) admitting $3^{n/3}$ cliques, and replacing each edge with a semaphore gadget (right). All initial cliques now correspond to $\mathcal{TC}^{\circlearrowleft}$ components, which are maximal when including semaphore vertices.

Outside of the trivial upper bound of 2^n, a better upper bound can be obtained as follows. Consider the following adaptation of the transitive closure: instead of adding an arc between vertices u and v when a journey exists, add an arc when a round trip exists. (Detecting round trips between all pairs of vertices is easily done adapting the temporal Dijkstra algorithm from [46].) By definition, $\mathcal{TC}^{\circlearrowleft}$ components of $\mathcal{G}$ correspond to bidirectional cliques in this adapted transitive closure of $\mathcal{G}$. Further modify the transitive closure as follows. For all arcs (u, v) such that (v, u) exists as well, replace both arcs by one (undirected) edge $\{u, v\}$. All other arcs are removed. The corresponding graph is undirected and cliques in this graph correspond to $\mathcal{TC}^{\circlearrowleft}$ components. By Moon and Moser [41], at most $3^{n/3}$ cliques can exist in undirected graphs, which implies the same upper bound $3^{n/3} < 2^{0.53n}$ holds for $\mathcal{TC}^{\circlearrowleft}$ components in temporal graphs. ◀

▶ **Theorem 47.** *$\mathcal{TC}^{\circlearrowleft}$ Component is solvable in time $O(n^2(m \log T + n \log n) + \min(n^k k^2, 2^{0.25n}))$.*

Proof. We start by constructing a modified version of the transitive closure H, where an arc (u, v) represents $u \circlearrowleft v$ instead of $u \rightsquigarrow v$. Initialize H with vertex set V. For all pairs of vertices, of which there are $O(n^2)$, do the following. Apply the temporal Dijkstra algorithm from [46] starting from u to obtain the smallest label at which v is reached from u through a journey, suppose label t. If v is unreachable from u, suppose $t = \infty$. This takes time $O(m \log T + n \log n)$. Then, if $t \neq \infty$, on temporal graph $\mathcal{G}_{[t,T]}$, apply again the temporal Dijkstra algorithm to check if u is reachable from v. This again takes time $O(m \log T + n \log n)$. If $t \neq \infty$ and if u is reachable from v in $\mathcal{G}_{[t,T]}$, then add arc (u, v) to H. Apply the same process in the other direction to check if $v \circlearrowleft u$. After having done this for all pairs of vertices, modify H as follows. For all arcs (u, v) such that (v, u) exists as well, replace both arcs by an (undirected) edge $\{u, v\}$. Remove all other arcs. H is now an undirected graph in which cliques correspond to $\mathcal{TC}^{\circlearrowleft}$ components. Since we're interested in cliques of size at least k, the brute force algorithm of testing each subset of size k runs in time $O(n^k k^2)$. Also, one can obtain a maximum-size clique in time $O(2^{n/4})$ [43]. All together, $\mathcal{TC}^{\circlearrowleft}$ Component can thus be solved in time $O(n^2(m \log T + n \log n) + \min(n^k k^2, 2^{0.25n}))$. ◀

▶ **Theorem 48.** *$\mathcal{TC}^{\circlearrowleft}$ Component is solvable in polynomial time $O(n + m)$ on temporal graphs with constant lifetimes $T \leq 2$.*

Proof. The case of $T = 1$ trivially reduces to finding connected components in the snapshot, doable in time $O(n + m)$.

For the case of $T = 2$, note that three possible $\mathcal{TC}^{\circlearrowleft}$ components can exist. The first (*resp.* second) type of $\mathcal{TC}^{\circlearrowleft}$ components are those which use only labels 1 (*resp.* 2). These $\mathcal{TC}^{\circlearrowleft}$ components correspond to connected components in the corresponding snapshot, and can thus be computed in time $O(n + m)$ as well. The third, and final, type of $\mathcal{TC}^{\circlearrowleft}$ components use both labels 1 and 2. We continue by proving such $\mathcal{TC}^{\circlearrowleft}$ components do not exist, finishing the proof.

By contradiction, let us suppose one such a $\mathcal{TC}^{\circlearrowleft}$ component exists, say V'. For any two vertices $u, v \in V'$, if a journey using only labels 1 (*resp.* 2) does not exist between u and v, then a journey using only labels 2 (*resp.* 1) must exist between u and v (or V' would not be a $\mathcal{TC}^{\circlearrowleft}$ component). Let us simplify this as a graph G with a copy of V' in which an edge with label i is drawn between vertices u and v if the corresponding journey between u and v uses only labels i. G is thus a complete graph with some arbitrary labelling of one label per edge, being either 1 or 2. We finish by showing that in G there is necessarily a spanning structure using only label 1, or using only label 2, implying that in fact V' is not a $\mathcal{TC}^{\circlearrowleft}$ component needing both labels 1 and 2, as a spanning structure using only labels 1 or only labels 2 exists which is sufficient for $\mathcal{TC}^{\circlearrowleft}$.

G contains a spanning structure using only labels 1 or only labels 2: We prove this property by induction. A K_2 graph trivially admits such a spanning structure. Now, if a K_n graph has this property, suppose *w.l.o.g.* that the spanning structure has only labels 1. Adding a vertex so as to obtain a K_{n+1} implies adding n edges connecting this vertex to the other vertices. To avoid having this property in the newly constructed K_{n+1}, none of these edges can have label 1 (since otherwise a spanning structure with only labels 1 exists), but they cannot all have label 2 either (since otherwise a spanning structure with only labels 2 exists), which is impossible. ◀

▶ **Theorem 49.** *$\mathcal{TC}^{\circlearrowleft}$ Component is NP-complete, for all constant lifetimes $T > 2$.*

Proof. We show how to reduce $\frac{n}{3}$ Clique to $\mathcal{TC}^{\circlearrowleft}$ Component in polynomial time. Given a graph $G = (V, E)$ for the $\frac{n}{3}$ Clique problem, create temporal graph $\mathcal{G}$ with (initially) the same vertex set V. For each edge $\{u, v\} \in E$, add a semaphore gadget between u and v in $\mathcal{G}$, with semaphore "intermediary" vertices uv and vu. On edges $\{u, uv\}$ and $\{v, vu\}$ place labels 1 and 3, and on edges $\{u, vu\}$ and $\{v, uv\}$ place labels 2.

We prove a clique of size $k = \frac{n}{3}$ exists in G if and only if a $\mathcal{TC}^{\circlearrowleft}$ component of size k^2 exists in $\mathcal{G}$.

$\frac{n}{3}$ Clique $\implies$ $\mathcal{TC}^{\circlearrowleft}$ Component

If a clique of size k exists in G of vertices $K = \{v_1, v_2, ..., v_k\}$, then consider the corresponding vertices in $\mathcal{G}$, as well as intermediary vertices $\{v_iv_j : v_i, v_j \in K, v_i \neq v_j\}$. Let's refer to this set of vertices in $\mathcal{G}$ as V', which is of size $k + 2\frac{k(k-1)}{2} = k^2$. V' is a $\mathcal{TC}^{\circlearrowleft}$ component, since all vertices can reach each other and then reach back:

- a vertex v_i can reach all other vertices v_j and back through journey $(v_i, v_iv_j, v_j, v_iv_j, v_i)$ using time steps $(1, 2, 2, 3)$ respectively;
- a vertex v_i can reach all intermediary vertices v_jv_k and back through journey $(v_i, v_iv_k, v_k, v_jv_k, v_k, v_iv_k, v_i)$ with time steps $(1, 2, 2, 2, 2, 3)$ respectively;
- an intermediary vertex v_iv_j can reach all vertices v_k and back through journey $(v_iv_j, v_i, v_iv_k, v_k, v_iv_k, v_i, v_iv_j)$ using time steps $(1, 1, 2, 2, 3, 3)$ respectively;
- an intermediary vertex v_iv_j can reach all other intermediary vertices v_kv_ℓ and back through journey $(v_iv_j, v_i, v_iv_\ell, v_\ell, v_kv_\ell, v_\ell, v_iv_\ell, v_i, v_iv_j)$ using time steps $(1, 1, 2, 2, 2, 2, 3, 3)$ respectively.

$\mathcal{TC}^{\circlearrowleft}$ Component $\implies$ $\frac{n}{3}$ Clique

If a $\mathcal{TC}^{\circlearrowleft}$ component V' of size $k^2 = \frac{n^2}{9}$ exists in $\mathcal{G}$, then consider all vertices $K = V' \cap V$. Trivially $|K| > 0$, furthermore $|K| > 1$ since otherwise the lone vertex $v \in K$ is part of a $\mathcal{TC}^{\circlearrowleft}$ component of size at most $2\Delta + 1$ (including all adjacent intermediary vertices), which leads to the asymptotic contradiction $k^2 = \frac{n^2}{9} \leq 2\Delta + 1 < 2n + 1$. We prove that K forms a clique of size k in G:

- for all pairs of vertices $v_i, v_j \in K$, intermediary vertex v_iv_j (*resp.* v_jv_i) must be included in V', since otherwise no round trip $v_i \circlearrowleft v_j$ (*resp.* $v_j \circlearrowleft v_i$) exists in V';
- for all intermediary vertices $v_kv_j : v_j \in K, v_k \notin K$ (thus having label 2), round trip $v_kv_j \circlearrowleft v_i$, where $v_i \in K$ isn't possible without v_k, *i.e.*, vertices v_kv_j cannot be included in V';
- for all intermediary vertices $v_jv_k : v_j \in K, v_k \notin K$ (thus having labels 1 and 3), round trip $v_i \circlearrowleft v_jv_k$, where $v_i \in K$ isn't possible without v_k, *i.e.*, vertices v_jv_k cannot be included in V';
- since V' is composed solely of vertices K and all their corresponding intermediary vertices, we have that $|V'| = k^2 = |K| + 2\frac{|K|(|K|-1)}{2} = |K|^2$, leading to $|K| = k$;
- finally, since all intermediary vertices of K are present in V' which can only be created through a semaphore gadget, which in turn can only be created if a corresponding edge is present in G, K forms a clique in G.

◀

▶ **Theorem 50.** *The worst-case number of closed $\mathcal{TC}^{\circlearrowleft}$ components is at least $2^{0.52\sqrt{n}}$, and at most $2^{0.53n}$.*

Proof. The lower bound comes from the fact that all $\mathcal{TC}^{\circlearrowleft}$ components in the construction from Theorem 46 are closed, and the upper bound transfers directly from Theorem 46. ◀

▶ **Lemma 65.** *Verifying if a vertex subset V' of size k is a solution for* Closed $\mathcal{TC}^{\circlearrowleft}$ Component *takes time $O(k^2(\min(m, k^2)\log T + k\log k))$.*

Proof. Construct a modified transitive closure H in the following manner. In $\mathcal{G}[V']$, apply for all pairs of vertices u and v, of which there are $O(k^2)$, the temporal Dijkstra algorithm from [46], starting from vertex u to obtain the smallest label t at which u reaches v. If u cannot reach v, then V' is not a solution vertex set. This can be done in time $O(\min(m, k^2)\log T + k\log k)$. Then, check if v can reach u in temporal graph $\mathcal{G}[V']_{[t,T]}$, doable again in time $O(\min(m, k^2)\log T + k\log k)$. If v cannot reach u, then V' is not a solution vertex set. Else, it means $u \circlearrowleft v$ in $\mathcal{G}[V']$. Apply the same process in the other direction to check if $v \circlearrowleft u$. When this is checked for all pairs of vertices, then V' is a solution vertex set for Closed $\mathcal{TC}^{\circlearrowleft}$ Component. In total, this thus takes time $O(k^2(\min(m, k^2)\log T + k\log k))$. ◀

▶ **Lemma 66.** *The existence of a closed $\mathcal{TC}^{\circlearrowleft}$ component of size $> k$ does not necessarily imply the existence of a (non-maximal) closed $\mathcal{TC}^{\circlearrowleft}$ component of size k.*

Proof. Consider Figure 15, in which an infinite family of temporal graphs is presented. For each of these graphs, say $\mathcal{G}$, a closed $\mathcal{TC}^{\circlearrowleft}$ component exists of size n. However, no (non-maximal) closed $\mathcal{TC}^{\circlearrowleft}$ component exists of size $n-1$, since considering any such set of vertices $V \setminus u_i$ (*resp.* v_i) would result in v_i (*resp.* u_i) not being able to reach any vertex v_j (*resp.* u_j) and back.

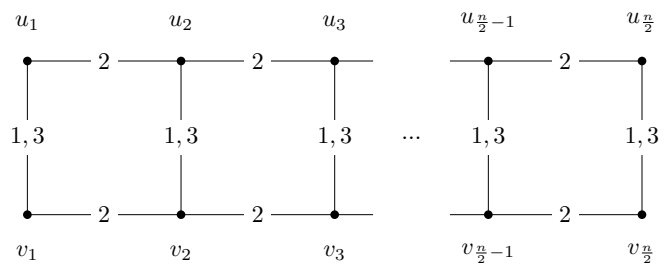


Figure 15 Temporal graph family in which a closed $\mathcal{TC}^{\circlearrowleft}$ component of size n exists but no non-maximal component of size $n-1$ exists.

◀

▶ **Theorem 51.** Closed $\mathcal{TC}^{\circlearrowleft}$ Component *is solvable in time $O(2^n(n^2(m\log T + n\log n))$.*

Proof. We present the brute force algorithm. Due to Theorem 66, we know it may not be sufficient to verify only the subsets of size exactly k; we may need to verify all subsets of size at least k, of which there exist $O(2^n)$. By Theorem 65, verifying such a subset requires $O(n^2(m\log T + n\log n))$ time. In total, the brute force algorithm thus takes time $O(2^n(n^2(m\log T + n\log n))$. ◀

▶ **Theorem 52.** *Closed $\mathcal{TC}^{\circlearrowleft}$ Component is solvable in polynomial time $O(n+m)$ on temporal graphs with constant lifetimes $T \leq 2$.*

Proof. In Theorem 48, we prove that all $\mathcal{TC}^{\circlearrowleft}$ components in temporal graphs of lifetime $T \leq 2$ correspond to connected components in $\mathcal{G}_1$ or $\mathcal{G}_2$. Connected components are necessarily closed, thus all $\mathcal{TC}^{\circlearrowleft}$ components in temporal graphs of lifetime $T \leq 2$ are necessarily closed. Since $\mathcal{TC}^{\circlearrowleft}$ Component can be solved in time $O(m+n)$ on such temporal graphs, Closed $\mathcal{TC}^{\circlearrowleft}$ Component can be solved in time $O(m+n)$ as well. ◀

▶ **Theorem 53.** *Closed $\mathcal{TC}^{\circlearrowleft}$ Component is NP-complete, for all constant lifetimes $T > 2$.*

Proof. By Theorem 65, a vertex subset can be verified to be a solution for Closed $\mathcal{TC}^{\circlearrowleft}$ Component in time $O(k^2(\min(m, k^2)\log T + k\log k))$. Closed $\mathcal{TC}^{\circlearrowleft}$ Component is thus in NP.

In the reduction used in Theorem 49, all $\mathcal{TC}^{\circlearrowleft}$ components in the transformed instance are closed. This suffices to prove that Closed $\mathcal{TC}^{\circlearrowleft}$ Component is NP-hard, for any constant lifetime $T > 2$. ◀