On Computing Large Temporal (Unilateral) Connected Components

Isnard Lopes Costa[2], Raul Lopes[3,4], Andrea Marino[1], and Ana Silva[1,2](✉)

[1] Dipartimento di Statistica, Informatica, Applicazioni, Università degli Studi di Firenze, Firenze, Italy
andrea.marino@unifi.it

[2] Departamento de Matemática, Universidade Federal do Ceará, Fortaleza, CE, Brazil
isnard.lopes@alu.ufc.br, anasilva@mat.ufc.br

[3] Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

[4] DIENS, Ecole normale supérieure de Paris, CNRS, Paris, France
raul.lopes@ens.psl.eu

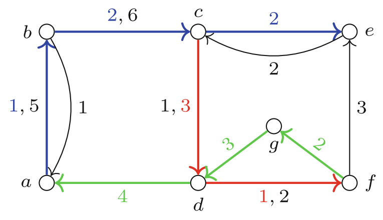
Abstract. A temporal (directed) graph is a graph whose edges are available only at specific times during its lifetime, τ. Paths are sequences of adjacent edges whose appearing times are either strictly increasing or non-strictly increasing (i.e., non-decreasing) depending on the scenario. Then, the classical concept of connected components and also of unilateral connected components in static graphs naturally extends to temporal graphs. In this paper, we answer the following fundamental questions in temporal graphs. (i) What is the complexity of deciding the existence of a component of size k, parameterized by τ, by k, and by $k+\tau$? We show that this question has a different answer depending on the considered definition of component and whether the temporal graph is directed or undirected. (ii) What is the minimum running time required to check whether a subset of vertices are pairwise reachable? A quadratic algorithm is known but, contrary to the static case, we show that a better running time is unlikely unless SETH fails. (iii) Is it possible to verify whether a subset of vertices is a component in polynomial time? We show that depending on the definition of component this test is NP-complete.

1 Introduction

A *(directed) temporal graph* (G, λ) *with lifetime* τ consists of a (directed) graph G together with a *time-function* $\lambda : E(G) \to 2^{[\tau]}$ which tells when each edge $e \in E(G)$ is available along the discrete time interval $[\tau]$. Given $i \in [\tau]$, the

(Partially) supported by: FUNCAP MLC-0191-00056.01.00/22 and PNE-0112-00061.01.00/16, CNPq 303803/2020-7, Italian PNRR CN4 Centro Nazionale per la Mobilità Sostenibile, and the group Casino/ENS Chair on Algorithmics and Machine Learning. Thanks to Giulia Punzi and Mamadou Kanté for interesting discussions. A full version of this paper is available at: https://arxiv.org/abs/2302.12068.

S.-Y. Hsieh et al. (Eds.): IWOCA 2023, LNCS 13889, pp. 282–293, 2023.
https://doi.org/10.1007/978-3-031-34347-6_24



$A' = \{a, b\}$ is a closed connected set, as a and b reach each other without using external vertices.

$A = \{a, b, c, d\}$ is a maximal closed connected set, i.e. a CLOSED TCC.

$B = \{a, b, c, d, e\}$ is a CLOSED TUCC but not a CLOSED TCC as, using only vertices in B, a, b, c, d reach each other, e reaches all the vertices in B and vice versa, except for d, which does not reach e. B is also a TCC, as d can reach e using the external vertex f.

$C = \{a, b, c, d, e, f\}$ is a TUCC as B forms a CLOSED TUCC, f is able to reach every other vertex directly or via the external vertex g. However, C is not a TCC as a, b, c, e cannot reach f.

Fig. 1. On the left a temporal graph, where on each edge e we depict $\lambda(e)$. Some of its components according to the non-strict model are reported on the right.

snapshot G_i refers to the subgraph of G containing exactly the edges available in time i. Temporal graphs, also appearing in the literature under different names, have attracted a lot of attention in the past decade, as many works have extended classical notions of Graph Theory to temporal graphs (we refer the reader to the survey [11] and the seminal paper [10]).

A crucial characteristic of temporal graphs is that a u, v-walk/path in G is valid only if it traverses a sequence of adjacent edges $e_1, \ldots, e_k$ at non-decreasing times $t_1 \leq \ldots \leq t_k$, respectively, with $t_i \in \lambda(e_i)$ for every $i \in [k]$. Similarly, one can consider strictly increasing sequences, i.e. with $t_1 < \ldots < t_k$. The former model is referred to as *non-strict* model, while the latter as *strict*. In both settings, we call such sequence a *temporal* u, v*-walk/path*, and we say that u *reaches* v. For instance, in Fig. 1, both blue and green paths are valid in the non-strict model, but only the green one is valid in the strict model, as the blue one traverses two edges with label 2. The red path is not valid in either model.

The non-strict model is more appropriate in situations where the time granularity is relatively big. This is the case in a disease-spreading scenario [19], where the spreading speed might be unclear or in "time-varying graphs", as in [14], where a single snapshot corresponds to all the edges available in a time interval, e.g. the set of all the streets available in a day. As for the strict model, it can represent the connections of the public transportation network of a city which are available only at precise scheduled times. All in all, there is a rich literature on both models and this is why we explore both settings.

Connected Sets and Components. Given a temporal graph $\mathcal{G} = (G, \lambda)$, we say that $X \subseteq V(G)$ is a *temporal connected set* if u reaches v *and* v reaches u, for every $u, v \in X$. Extending the classical notion of connected components in static graphs, in [2] the authors define a *temporal connected component* (TCC for short) as a maximal connected set of $\mathcal{G}$. Such constraint can be strengthened to the existence of such paths using only vertices of X. Formally, X is a *closed temporal connected component* (CLOSED TCC for short) if, for every $u, v \in X$, we have that u reaches v *and* v reaches u through temporal paths contained in X. See Fig. 1 for an example of TCC and CLOSED TCC.

Unilateral Connected Components. In the same fashion, also the concept of *unilateral connected components* can be extended to temporal graphs. In static graph theory, they are a well-studied relaxation of connected components which asks for a path from u to v <u>or</u> vice versa, for every pair u, v in the component [1,4]. More formally, in a directed graph G, we say that $X \subseteq V(G)$ is a *unilateral connected set* if u reaches v <u>or</u> v reaches u, for every $u, v \in X$. X is a *unilateral connected component* if it is maximal. In this paper, we introduce the definition of a *(closed) unilateral temporal connected set/component*, which can be seen as the immediate translation of unilateral connected component to the temporal context. Formally, $X \subseteq V(G)$ is a *temporal unilateral connected set* if u reaches v *or* v reaches u, for every $u, v \in X$, and it is a *closed unilateral connected set* if this holds using paths contained in X. Finally, a *(closed) temporal unilateral connected component* ((CLOSED) TUCC for short) is a maximal (closed) temporal unilateral connected set. See again Fig. 1 for an example.

Problems. In this paper, we deal with four different definitions of temporal connected components, depending on whether they are unilateral or not, and whether they are closed or not. In what follows, we pose three questions, and we comment on partial knowledge about each of them. Later on, we discuss our results, which close almost all the gaps found in the literature. We start by asking the following.

Question 1 (Parameterized complexity). What is the complexity of deciding the existence of temporal components of size at least k parameterized by *(i)* τ, i.e. the lifetime, *(ii)* k, and *(iii)* $k + \tau$?

In order to answer Question 1 for the strict model, there is a very simple parameterized reduction from k-clique, known to be W[1]-hard when parameterized by k [7], to deciding the existence of connected components (both closed or not and both unilateral or not) of size at least k in undirected temporal graphs. This reduction has appeared in [5]. Given an undirected graph G, we can simply consider the temporal graph $\mathcal{G} = (G, \lambda)$ where $\lambda(uv) = \{1\}$ for all $uv \in E(G)$ (i.e., $\mathcal{G}$ is equal to G itself). As u temporally reaches v if and only if $uv \in E(G)$, one can see that all those problems are now equivalent to deciding the existence of a k-clique in G. Observe that we get W[1]-hardness when parameterized by k or $k + \tau$, and para-NP-completeness when parameterized by τ, both in the undirected and the directed case.[1] However, this reduction does not work in the case of the *non-strict* model, leaving Question 1 open. Indeed the reductions in [2] and in [6] for (CLOSED) TCCs, which work indistinctly for both the strict or the non-strict models, are not parameterized reductions. We also observe that the aforementioned reductions work on the non-strict model only for $\tau \geq 4$.

Another question of interest is the following. Letting n be the number of vertices in $\mathcal{G}$ and M be the number of *temporal edges*,[2] it is known that, in

[1] In the directed case, it suffices to replace each edge of the input graph with two opposite directed edges between the same endpoints.

[2] $M = \sum_{e \in E(\mathcal{G})} |\lambda(e)|$.

Table 1. A summary of our results for the parameterized complexity of computing components of size at least k of a temporal graph $\mathcal{G}$ having lifetime τ in the *non-strict* model. "W[1]-h" stands for W[1]-hardness and "p-NP" stands for para-NP-completeness. For the strict model the entries are W[1]-h in the third and fourth columns and p-NP in the second one already for $\tau = 1$, both for the directed and the undirected case.

	Par. τ	Par. k	Par. $k + \tau$
TCC	p-NP $\tau \geq 2$ (Theorem 1)	W[1]-h Dir. $\tau \geq 2$ (Theorem 3) and Undir. (Theorem 2)	W[1]-h Dir. (Theorem 3) FPT Undir. (Theorem 5)
TUCC		W[1]-h Dir. $\tau \geq 2$ (Theorem 3) FPT Undir. (Theorem 5)	
CLOSED TCC		W[1]-h Dir. $\tau \geq 3$ (Theorem 4)	W[1]-h Dir. (Theorem 4) FPT Undir. (Theorem 5)
CLOSED TUCC		W[1]-h Dir. $\tau \geq 3$ (Theorem 4) FPT Undir. (Theorem 5)	

order to verify whether $X \subseteq V(G)$ is a connected set in $\mathcal{G}$, we can simply apply $O(n)$ single source "best" path computations (see e.g. [17]), resulting in a time complexity of $O(n \cdot M)$. This is $O(M^2)$ if $\mathcal{G}$ has no isolated vertices, a natural assumption when dealing with connectivity problems. As in static graphs testing connectivity can be done in linear time [8], we ask whether the described algorithm can be improved.

Question 2 (Lower bound on checking connectivity). Given a temporal graph $\mathcal{G}$ and a subset $X \subseteq V(\mathcal{G})$, what is the minimum running time required to check whether X is a (unilateral) connected set?

Finally we focus on one last question.

Question 3 (Checking maximality). Given a temporal graph $\mathcal{G}$ and a subset $X \subseteq V(\mathcal{G})$, is it possible to verify, in polynomial time, whether X is a component, i.e. a maximal (closed) (unilateral) connected set?

For Question 3, we first observe that the property of being a temporal (unilateral) connected set is hereditary (forming an independence system [12]), meaning that every subset of a (unilateral) connected set is still a (unilateral) connected set. For instance, in Fig. 1, every subset of the connected set $B = \{a, b, c, d, e\}$ is a connected set. Also, checking whether $X' \subseteq V(G)$ is a temporal (unilateral) connected set can be done in time $O(n \cdot M)$, as discussed above. We can then check whether X is a maximal such set in time $O(n^2 \cdot M)$: it suffices to test, for every $v \in V(G) \backslash X$, whether by adding v to X we still get a temporal (unilateral) connected set. On the other hand, *closed* connected (unilateral) sets are not hereditary, because by removing vertices from the set we could destroy the paths between other members of the set. This is the case for the closed connected set $A = \{a, b, c, d\}$ in Fig. 1, since by removing d there are no temporal paths from c to a nor b using only vertices in the remainder of the set. This implies that the same approach as before does not work, i.e., we cannot check whether X is maximal by adding to X a *single* vertex at a time, then checking for connectivity.

For instance, the closed connected set $A' = \{a, b\}$ in Fig. 1 cannot be grown into the closed connected set A by adding one vertex at a time, since both $A' \cup \{c\}$ and $A' \cup \{d\}$ are not closed connected sets. Hence, the answer to Question 3 for closed sets does not seem easy, and until now was still open.

Our Results. Our results concerning Question 1 are reported in Table 1 for the non-strict model, since for the strict model all the entries would be W[1]-hard or para-NP-complete already for $\tau = 1$, as we argued before. In the non-strict model, we observe instead that the situation is much more granulated. If $\tau = 1$, then all the problems become the corresponding traditional ones in static graphs, which are all polynomial (see Paragraph "Related works"). As for bigger values of τ, the complexity depends on the definition of component being considered, and whether the temporal graph is directed or not. Table 1 considers $\tau > 1$, reporting on negative results, "$\tau \geq x$" for some x meaning that the negative result starts to hold for temporal graphs of lifetime at least x.

The second column of Table 1 addresses Question 1*(i)*, i.e., parameterization by τ. We prove that, for all the definitions of components being considered, the related problem becomes immediately para-NP-complete as soon as τ increases from 1 to 2; this is done in Theorem 1. This reduction improves upon the reduction of [2], which holds only for $\tau \geq 4$.

Question 1*(ii)* (parameterization by k) is addressed in the third column of Table 1. Considering first directed temporal graphs, we prove that all the problems are W[1]-hard. In particular, deciding the existence of a TCC or TUCC of size at least k is W[1]-hard already for $\tau \geq 2$ (Theorem 3). As for the existence of closed components, W[1]-hardness also holds as long as $\tau \geq 3$ (Theorem 4). Observe that, since τ is constant in both results, these also imply the W[1]-hardness results presented in the last column, thus answering also Question 1*(iii)* (parameterization by $k + \tau$) for directed graphs. On the other hand, if the temporal graph is undirected, then the situation is even more granulated. Deciding the existence of a TCC of size at least k remains W[1]-hard, but only if τ is unbounded. This is complemented by the answer to Question 1*(iii)*, presented in the last column of Table 1: TCC and (even) CLOSED TCC are FPT on undirected graphs when parameterized by $k + \tau$ (Theorem 5). We also give FPT algorithms when parameterized by k for unilateral components, namely TUCC and CLOSED TUCC. Observe how this differs from TCC, whose corresponding problem is W[1]-hard, meaning that unilateral and traditional components behave very differently when parameterized by k.

In summary, Table 1 answers Question 1 for almost all the definitions of components, both for directed and undirected temporal graphs, leaving open only the problems of, given an undirected temporal graph, deciding the existence of a CLOSED TCC of size at least k when parameterized by k, and solving the same problem for CLOSED TCC and CLOSED TUCC in directed temporal graphs where $\tau = 2$.

Concerning Questions 2 and 3, our results are summarized in Table 2. All these results hold both for the strict and the non-strict models. For Question 2, we prove that the trivial $O(M^2)$ algorithm to test whether S is a (closed)

Table 2. Our results for Question 2 and Question 3, holding for both the *strict* and the *non-strict* models. Recall that a component is a (inclusion-wise) maximal connected set. The $O(\cdot)$ result is easy and explained in the introduction. M (resp. n) denotes the number of temporal edges (resp. nodes) in $\mathcal{G}$.

	CHECK WHETHER $X \subseteq V$ IS A CONNECTED SET	CHECK WHETHER $X \subseteq V$ IS A COMPONENT
TCC	$\Theta(M^2)$ (Theorem 6)	$O(n^2 \cdot M)$
TUCC		
CLOSED TCC		NP-c (Theorem 7)
CLOSED TUCC		

(unilateral) connected set is best possible, unless the Strong Exponential Time Hypothesis (SETH) fails [9]. For Question 3, in the case of TCC and TUCC, we have already seen that checking whether a set $X \subseteq V$ is a component can be done in $O(n^2 \cdot M)$. Interestingly, for CLOSED TCC and CLOSED TUCC, we answer negatively (unless $\mathsf{P} = \mathsf{NP}$) to Question 3.

Related Work. The known reductions for temporal connected components in the literature [2,6] which considers the non-strict setting are not parameterized and leave open the case when $\tau = 2$ or 3. The reductions we give here are parameterized (Theorems 3 and 4) and Theorem 1 closes also the cases $\tau = 2$ and 3. Furthermore, in [6] they show a series of interesting transformations but none of them allows us to apply known negative results for the strict model to the non-strict one. There are many other papers about temporal connected components in the literature, including [5], where they give an example where there can be an exponential number of temporal connected components in the strict model. In [14], the authors show that the problem of computing TCCs is a particular case of finding cliques in the so-called *affine graph*. This does not imply that the problem is NP-complete as claimed, since in order to prove hardness one needs to do a reduction on the opposite direction, i.e., a reduction from a hard problem to TCC instead. Finally, we remark that there are many results in the literature concerning unilateral components in static graphs [1], also with applications to community detection [13]. Even though the number of unilateral components in a graph is exponential [1], deciding whether there is one of size at least k is polynomial.

2 Parameterized Complexity Results

Parameterization by τ. We start by proving the result in the first column of Table 1, about para-NP-completeness wrt. to the lifetime τ, which applies to all the definitions of temporal components. For (CLOSED) TCC we present a reduction from the NP-complete problem MAXIMUM EDGE BICLIQUE (MEBP for short) [16]. A *biclique* in a graph G is a complete bipartite subgraph of G. The MEBP problem consists of, given a bipartite graph G and an integer k, deciding whether G has a biclique with at least k edges. Using the same construction, we prove hardness of (CLOSED) TUCC reducing from the NP-complete problem $2K_2$-FREE EDGE SUBGRAPH [18]. In this problem we are given a bipartite graph

G and an integer k, and are trying to decide whether G has a $2K_2$-free subgraph with at least k edges.

The main idea of the reductions is to generate a temporal graph $\mathcal{G}$ whose underlying graph is the line graph L of a bipartite graph H with parts X, Y. Recall that, for each $u \in X \cup Y$, there is a clique in L formed by all the edges incident to u; denote such clique by C_u. We make active in timestep 1 the edges within C_u for every $u \in X$, and in timestep 2 the edges within C_u for every $u \in Y$. Doing so, we ensure that any pair of vertices of $\mathcal{G}$ associated with a biclique in H reach one another in $\mathcal{G}$. We prove that there exists a biclique in H with at least k edges if and only if there exists a CLOSED TCC in $\mathcal{G}$ of size at least k. The result extends to TCCs, as every TCC is also a CLOSED TCC. For the unilateral case, we can relax the biclique to a $2K_2$-free graph since only one reachability relation is needed. As a result, we get the following.

Theorem 1. *For every fixed $\tau \geq 2$ and given a temporal graph $\mathcal{G} = (G, \lambda)$ of lifetime τ and an integer k, it is* NP*-complete to decide if $\mathcal{G}$ has a* (CLOSED) TCC *or a* (CLOSED) TUCC *of size at least k, even if G is the line graph of a bipartite graph.*

W*[1]-hardness by k.* We now focus on proving the W[1]-hardness results in the second column of Table 1 concerning parameterization by k, which also imply some of the results of the third column. The following W[1]-hardness results (Theorem 2, 4, and 3) are parameterized reductions from k-CLIQUE. The general objective is constructing a temporal graph $\mathcal{G}$ in a way that vertices in $\mathcal{G}$ are in the same component if and only if the corresponding nodes in the original graph are adjacent. Notice that we have to do this while: (i) ensuring that the size of the desired component is $f(k)$ for some computable function k (i.e., this is a parameterized reduction); and (ii) avoiding that the closed neighborhood of a vertex forms a component, so as to not a have a false "yes" answer to k-CLIQUE. To address these tasks, we rely on different techniques. The first reduction concerns TCC in undirected graphs and requires τ to be unbounded, as for τ bounded we show that the problem is FPT by $k + \tau$ (Theorem 5). The technique used is a parameterized evolution of the so-called *semaphore* technique used in [2,6], which in general replaces edges by labeled diamonds to control paths of the original graph. However, while the original reduction gives labels in order to ensure that paths longer than one are broken, the following one allows the existence of paths longer than one. But if a temporal path from u to v exists for $uv \notin E(G)$, then the construction ensures the non-existence of temporal paths from v to u. Because of this property, the reduction does not extend to TUCCs, which we prove to be FPT when parameterized by k instead (Theorem 5).

Theorem 2. *Given a temporal graph $\mathcal{G}$ and an integer k, deciding if $\mathcal{G}$ has a* TCC *of size at least k is* W[1]*-hard with parameter k.*

Proof. We make a parameterized reduction from k-CLIQUE. Let G be graph and $k \geq 3$ be an integer. We construct the temporal graph $\mathcal{G} = (G', \lambda)$ as follows.

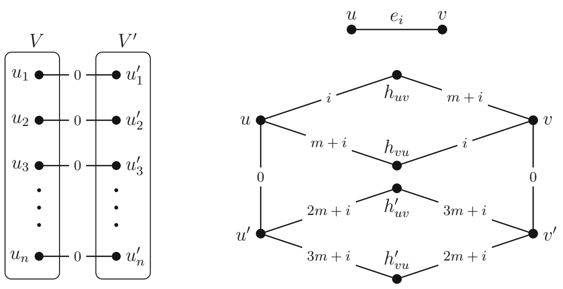


Fig. 2. Construction used in the proof of Theorem 2. On the left, the two copies of $V(G)$ and the edges between them, active in timestep 0. On the right, the edge $e_i \in E(G)$ and the associated gadget in $\mathcal{G}$.

See Fig. 2 to follow the construction. First, add to G' every vertex in $V(G)$ and make $V = V(G)$. Second, add to G' a copy u' of every vertex $u \in V$ and define $V' = \{u' \mid u \in V\}$. Third, for every pair u, u' with $u \in V$ and $u' \in V'$ add the edge uu' to G' and make all such edges active at timestep 0. Fourth, consider an arbitrary ordering $e_1, \ldots, e_m$ of the edges of G and, for each edge $e_i = uv$, create four new vertices $\{h_{uv}, h_{vu}, h'_{uv}, h'_{vu} \mid uv \in E(G)\}$, adding edges:

- uh_{uv} and vh_{vu}, active at time i;
- $u'h'_{uv}$ and $v'h'_{vu}$, active at time $2m+i$;
- $h_{vu}u$ and $h_{uv}v$, active at time $m+i$; and
- $h'_{vu}u'$ and $h'_{uv}v'$, active at time $3m+i$.

Denote the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$ by H, and the set $\{h'_{uv}, h'_{vu} \mid uv \in E(G)\}$ by H'. We now prove that G has a clique of size at least k if and only if $\mathcal{G}$ has a TCC of size at least $2k$. Given a clique C in G, it is easy to check that $C \cup \{u \in V' \mid u \in C\}$ is a TCC.

Now, let $S \subseteq V(G')$ be a TCC of $\mathcal{G}$ of size at least $2k$. We want to show that either $C = \{u \in V(G) \mid u \in S \cap V\}$ or $C' = \{u \in V(G) \mid u' \in S \cap V'\}$ is a clique of G of size at least k. This part of the proof combines a series of useful facts, which we cannot include here due to space constraints. In what follows we present a sketch of it.

First, we argue that both C and C' are cliques in G. Then, by observing that the only edges between $V \cup H$ and $V' \cup H'$ are those incident to V and V' at timestep 0, we conclude that either $S \subseteq V \cup H$ or $S \subseteq V' \cup H'$. Since the cases are similar, we assume the former. If $|S \cap V| \geq k$, then C contains a clique of size at least k and the result follows. Otherwise, we define $E_S = \{uv \in E(G) \mid \{h_{uv}, h_{vu}\} \cap S \neq \emptyset\}$. That is, E_S is the set of edges of G related to vertices in $S \cap H$. We then prove the following claim.

Claim. Let $a, b \in S \cap H$ be associated with distinct edges g, g' of G sharing an endpoint v. If u and w are the other endpoints of g and g', respectively, then u and w are also adjacent in G. Additionally, either $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, or $|S \cap H| \leq 2$.

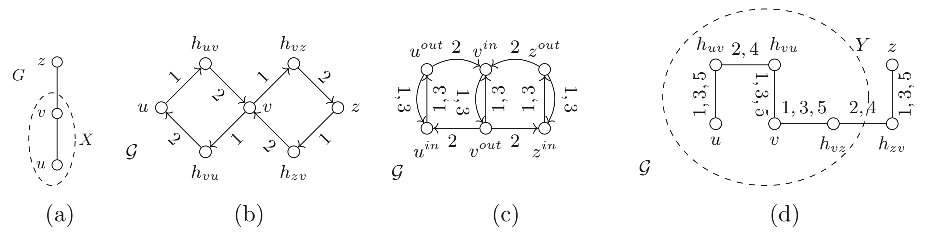


Fig. 3. Examples for some of our reductions. Given the graph in (a), Theorem 3 constructs the directed temporal graph in (b), Theorem 4 constructs the directed temporal graph in (c), and, given additionally set X in (a), Theorem 7 contructs the temporal graph $\mathcal{G}$ and set Y in (d).

To finish the proof, we first recall that we are in the case $|S \cap H| \geq k+1$. By our assumption that $k \geq 3$, note that the above claim gives us that $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, which in turn implies that $|E_S| = |S \cap H|$. Additionally, observe that, since $|S \cap H| \geq 4$, the same claim also gives us that there must exist $w \in V$ such that e is incident to w for every $e \in E_S$. Indeed, the only way that 3 distinct edges can be mutually adjacent without being all incident to the same vertex is if they form a triangle. Supposing that 3 edges in E_S form a triangle $T = (a, b, c)$, since $|E_S| \geq 4$, there exists an edge $e \in E_S \setminus E(T)$. But now, since G is a simple graph, e is incident to at most one between a, b and c, say a. We get a contradiction wrt. the aforementioned claim as in this case e is not incident to edge $bc \in E_S$. Finally, by letting $C'' = \{v_1, \ldots, v_k\}$ be any choice of k distinct vertices such that $\{wv_1, \ldots, wv_k\} \subseteq E_S$, our claim gives us that v_i and v_j are adjacent in G, for every $i, j \in [k]$; i.e., C'' is a k-clique in G. □

The following result concerns TCC and TUCC in directed temporal graphs. It is important to remark that for TCC and τ unbounded, we already know that the problem is W[1]-hard because of Theorem 2 which holds for undirected graphs and extends to directed ones. However, the following reduction applies specifically for directed ones already for $\tau = 2$. The technique used here is the previously mentioned semaphore technique, made parameterized by exploiting the direction of the edges. Namely, we reduce from k-Clique by replacing every edge uv of G by two vertices w_{uv} and w_{vu} and the directed temporal paths $(u, 1, w_{uv}, 2, v)$ and $(v, 1, w_{vw}, 2, u)$. See Fig. 3(b) to see the temporal graph obtained from the graph in Fig. 3(a). One can check that G has a clique of size at least k if and only if $\mathcal{G}$ has a TCC of size at least k. For TUCC, we only need to add one of w_{uv} or w_{vu}.

Theorem 3. *Given a directed temporal graph $\mathcal{G}$ and an integer k, deciding if $\mathcal{G}$ has a* TCC *of size at least k is* W[1]*-hard with parameter k, even if $\mathcal{G}$ has lifetime* 2. *The same holds for* TUCC.

The next result concerns closed TCCs and TUCCs. In this case, we also reduce from k-Clique, but we cannot apply the semaphore technique as before. Indeed, as we are dealing with closed components, nodes must be reachable using vertices inside the components, while the semaphore technique would make them reachable via additional nodes, which do not necessarily reach each other. For this reason, in the following we introduce a new technique subdividing nodes, instead of edges, in order to break paths of the original graph of length longer than one, being careful to allow that these additional nodes reach each other. The construction is shown in Fig. 3, which shows how to construct temporal graph $\mathcal{G}$ in Fig. 3(c), given graph G in Fig. 3(a) in a way that graph G has a clique of size k if and only if $\mathcal{G}$ has a CLOSED TCC (TUCC) of size at least $2k$.

Theorem 4. *Given a directed temporal graph $\mathcal{G}$ and an integer k, deciding if $\mathcal{G}$ has a* CLOSED TCC *of size at least k is* $\mathsf{W}[1]$*-hard with parameter k, even if $\mathcal{G}$ has lifetime* 3*. The same holds for* CLOSED TUCC.

FPT Algorithms. We now show our FPT algorithms to find (CLOSED) TCCs and (CLOSED) TUCCs in undirected temporal graphs, as for directed temporal graphs we have proved $\mathsf{W}[1]$-hardness. In particular, we prove the following result.

Theorem 5. *Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ, and a positive integer k, there are algorithms running in time*

1. *$O(k^{k\cdot\tau} \cdot n)$ that decides whether there is a* TCC *of size at least k;*
2. *$O(2^{k^\tau} \cdot n)$ that decides whether there is a* CLOSED TCC *of size at least k;*
3. *$O(k^{k^2} \cdot n)$ that decides whether there is a* TUCC *of size at least k; and*
4. *$O(2^{k^k} \cdot n)$ that decides whether there is a* CLOSED TUCC *of size at least k.*

Proof. The *reachability digraph* R associated to $\mathcal{G}$ is a directed graph with the same vertex set as $\mathcal{G}$, and such that uv is an edge in R if and only u reaches v in $\mathcal{G}$ and $u \neq v$. This is related to the *affine* graph in [14]. Observe that finding a TCC (resp. TUCC) in $\mathcal{G}$ of size at least k is equivalent to finding a set $S \subseteq V(\mathcal{G})$ in R of size *exactly* k such that $uv \in E(R)$ and (resp. or) $vu \in E(R)$ for every pair $u, v \in V(R)$. As for finding a CLOSED TCC (resp. CLOSED TUCC), we need to have the same property, except that all subsets of size *at least* k must be tested (recall that being a closed connected (unilateral) set is not hereditary). Therefore, if Δ is the maximum degree of R, then testing connectivity takes time $O(k^\Delta \cdot n)$ (it suffices to test all subsets of size $k-1$ in $N(u)$, for all $u \in V(R)$), while testing closed connectivity takes time $O(2^\Delta \cdot n)$ (it suffices to test all subsets of size *at least* $k-1$ in $N(u)$, for all $u \in V(R)$). The proofs then consist in bounding the value Δ in each case. □

It is important to observe that, for unilateral components, these bounds depend only on k, while for TCCs and CLOSED TCCs they depend on both k and τ. This is consistent with the fact that we have proved that for TCC the problem is $\mathsf{W}[1]$-hard when parameterized just by k (Theorem 2).

3 Checking Connectivity and Maximality

This section is focused on Questions 2 and 3. The former is open for all definitions of components for both the strict and the non-strict models. We answer to question providing the following conditional lower bound, which holds for both models, where the notation $\tilde{O}(\cdot)$ ignores polylog factors. We apply the technique used for instance in [3] to prove lower bounds for polynomial problems, which intuitively consists of an exponential reduction from SAT to our problem.

Theorem 6. *Consider a temporal graph $\mathcal{G}$ on M temporal edges. There is no algorithm running in time $\tilde{O}(M^{2-\epsilon})$, for some ϵ, that decides whether G is temporally (unilaterally) connected, unless* SETH *fails.*

We now focus on Question 3. We prove the results in the second column of Table 2, about the problem of deciding whether a subset of vertices Y of a temporal graph is a component, i.e. a maximal connected set. The question is open both for the strict and the non-strict model. We argued already in the introduction that this is polynomial for TCC and TUCC for both models. In the following we prove NP-completeness for CLOSED TCC and CLOSED TUCC on undirected graphs. The results extend to directed graphs as well.

Theorem 7. *Let $\mathcal{G}$ be a (directed) temporal graph, and $Y \subseteq V(\mathcal{G})$. Deciding whether Y is a* CLOSED TCC *is* NP*-complete. The same holds for* CLOSED TUCC.

Proof. We reduce from the problem of deciding whether a subset of vertices X of a given a graph G is a maximal 2-club, where a 2-club is a set of vertices C such that $G[C]$ has diameter at most 2. This problem has been shown to be NP-complete in [15]. Let us first focus on the strict model. In this case, given G we can build a temporal graph $\mathcal{G}$ with only two snapshots, both equal to G. Observe that X is a 2-club in G if and only if X is a CLOSED TCC in $\mathcal{G}$. Indeed, because we can take only one edge in each snapshot and $\tau = 2$, we get that temporal paths will always have length at most 2. This also extends to CLOSED TUCC by noting that all paths in $\mathcal{G}$ can be temporally traversed in both directions.

In the case of the non-strict model, the situation is more complicated as in each snapshot we can take an arbitrary number of edges resulting in paths arbitrarily long. We show the construction for CLOSED TCC in what follows. Let $\mathcal{G}$ be obtained from G by subdividing each edge uv of G twice, creating vertices h_{uv} and h_{vu}, with $\lambda(uh_{uv}) = \lambda(vh_{vu}) = \{1, 3, 5\}$, and $\lambda(h_{uv}h_{vu}) = \{2, 4\}$. See Fig. 3 (d) for an illustration.

Given (G, X), the instance of maximal 2-club, we prove that X is a maximal 2-club in G iff $Y = X \cup N_H(X)$ is a CLOSED TCC in $\mathcal{G}$. For this, it suffices to prove that, given $X' \subseteq V(G)$ and defining Y' similarly as before w.r.t. X', we have that $G[X']$ has diameter at most 2 iff Y' is a closed temporal connected set. The proof extends to CLOSED TUCC by proving that every CLOSED TCC is also a CLOSED TUCC and vice-versa. □

References

1. Arjomandi, E.: On finding all unilaterally connected components of a digraph. Inf. Process. Lett. **5**(1), 8–10 (1976)
2. Bhadra, S., Ferreira, A.: Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In: Pierre, S., Barbeau, M., Kranakis, E. (eds.) ADHOC-NOW 2003. LNCS, vol. 2865, pp. 259–270. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39611-6_23
3. Borassi, M., Crescenzi, P., Habib, M.: Into the square: on the complexity of some quadratic-time solvable problems. In: ICTCS. Electronic Notes in Theoretical Computer Science, vol. 322, pp. 51–67. Elsevier (2015)
4. Borodin, A.B., Munro, I.: Notes on efficient and optimal algorithms. Technical report, U. of Toronto and U. of Waterloo, Canada (1972)
5. Casteigts, A.: Finding structure in dynamic networks. arXiv preprint arXiv:1807.07801 (2018)
6. Casteigts, A., Corsini, T., Sarkar, W.: Simple, strict, proper, happy: a study of reachability in temporal graphs. arXiv preprint arXiv:2208.01720 (2022)
7. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness ii: on completeness for w [1]. Theor. Comput. Sci. **141**(1–2), 109–131 (1995)
8. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM **16**(6), 372–378 (1973)
9. Impagliazzo, R., Paturi, R.: On the complexity of k-sat. J. Comput. Syst. Sci. **62**(2), 367–375 (2001)
10. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. In: Yao, F.F., Luks, E.M., eds, Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, 21–23 May 2000. Portland, pp. 504–513. ACM (2000)
11. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. Soc. Netw. Anal. Min. **8**(1), 1–29 (2018). https://doi.org/10.1007/s13278-018-0537-7
12. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Generating all maximal independent sets: Np-hardness and polynomial-time algorithms. SIAM J. Comput. **9**(3), 558–565 (1980)
13. Levorato, V., Petermann, C.: Detection of communities in directed networks based on strongly p-connected components. In: 2011 International Conference on Computational Aspects of Social Networks (CASoN), pp. 211–216. IEEE (2011)
14. Nicosia, V., Tang, J., Musolesi, M., Russo, G., Mascolo, C., Latora, V.: Components in time-varying graphs. Chaos: Interdisc. J. Nonlinear Sci. **22**(2), 023101 (2012)
15. Foad Mahdavi Pajouh and Balabhaskar Balasundaram: On inclusionwise maximal and maximum cardinality k-clubs in graphs. Discret. Optim. **9**(2), 84–97 (2012)
16. Peeters, R.: The maximum edge biclique problem is np-complete. Discret. Appl. Math. **131**(3), 651–654 (2003)
17. Huanhuan, W., Cheng, J., Huang, S., Ke, Y., Yi, L., Yanyan, X.: Path problems in temporal graphs. Proc. VLDB Endowment **7**(9), 721–732 (2014)
18. Yannakakis, M.: Computing the minimum fill-in is np-complete. SIAM J. Algebraic Discrete Methods **2**(1), 77–79 (1981)
19. Zschoche, P., Fluschnik, T., Molter, H., Niedermeier, R.: The complexity of finding small separators in temporal graphs. J. Comput. Syst. Sci. **107**, 72–92 (2020)