Graph Algorithms (Spring 2026)

11. Temporal components and Temporal spanners

Teacher: Arnaud Casteigts *Assistant: Matteo De Francesco*

In this second class on temporal graphs, we discuss two topics that have received significant attention in recent years, namely *temporal components* and *temporal spanners*. One reason is that these concepts behave very differently in temporal graphs, compared to their static analogs.

11.0.1 Temporal components

Given a temporal graph $\mathcal{G} = (V, E, \lambda)$, a **temporal component** is a subset of vertices $V' \subseteq V$ such that for all $u, v \in V'$, u can reach v by a temporal path (shorthand, $u \rightsquigarrow v$). We have already seen that temporal reachability is *non-transitive*. An important consequence of this is that maximal temporal components may overlap, as illustrated in fig. 1.

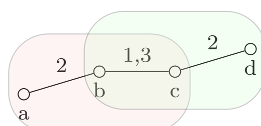


Figure 1: Two overlapping maximal components.

This feature is specific to temporal graphs. In static graphs (directed or not), maximal components do not overlap: they partition V into disjoint subsets. Unfortunately, this situation is a source of computational hardness for many problems. Let's consider the following two problems:

TEMPORAL COMPONENT
Input: a temporal graph $\mathcal{G}$ and an integer k.
Question: does $\mathcal{G}$ admit a temporal component of size k?

We can define analoguously the problem STRICT TEMPORAL COMPONENT where the component must be based on strict temporal paths. We will show that these two problems are NP-hard, using two reductions from the CLIQUE problem in static graphs. Let us start with the reduction for STRICT TEMPORAL COMPONENT, which is straightforward:

Theorem 11.1. STRICT TEMPORAL COMPONENT *is* NP-hard.

Proof. Given an instance for the CLIQUE problem, say, a graph $G = (V, E)$ and an integer k (where the question is whether G admits a clique of size k), one can construct a temporal graph $\mathcal{G}$ whose footprint is G itself and λ assigns a single and identical label to every edge (say, label 1). Because the temporal paths in $\mathcal{G}$ are required to be strict, two vertices can reach each other if and only if an edge exists between them. Thus, *a clique of size k exists in G if and only if a temporal component of size k exists in $\mathcal{G}$.* As CLIQUE is NP-hard, this implies that STRICT TEMPORAL COMPONENT is also NP-hard. □

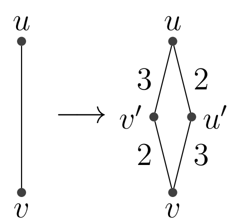
Pseudo-code:

```
has_clique(G, k):
   𝒢 ← temporal graph with footprint G and label 1 for every edge.
   return has_temporal_component(𝒢, k)
```

Clearly, the above construction does not work in the non-strict setting, because the constructed temporal graph would then form a single component. The reduction is slightly more complex:

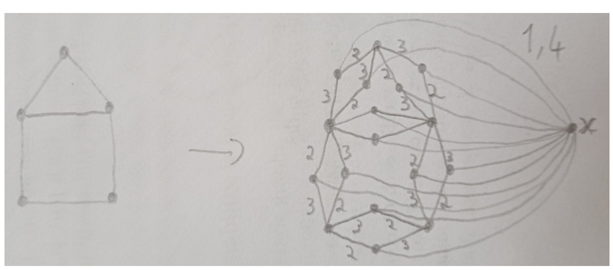
Theorem 11.2. TEMPORAL COMPONENT *is NP-hard.*

Proof. Given an instance $(G = (V, E), k)$ for the CLIQUE problem, one can build a temporal graph where every edge $\{u, v\} \in E$ is replaced with a *semaphore gadget*, i.e. an alternating cycle (u, u', v, v'), where u' and v' are extra intermediate vertices, and the labels along the cycle are respectively set to $2, 3, 2, 3$ (see the first picture). We then create another additional



(a) The semaphore gadget.

vertex x and add an edge between x and all the extra intermediate vertices, with labels 1 and 4. (See the second picture.)



As before, we obtain that two vertices u and v in V can reach each other in $\mathcal{G}$ if and only if they share an edge in G. Additionally, all of the $2|E|$ extra intermediate vertices can reach each other through x. These vertices can also reach (and be reached by) all the original vertices. Finally, observe that this construction does not create new reachability among the original vertices. Thus, *a clique of size k exists in G iff a temporal component of size $2 \cdot |E| + 1 + k$ exists in $\mathcal{G}$*, which implies that Temporal Component is NP-hard. □

Pseudo-code:

```
has_clique(G, k):
    𝒢 ← temporal graph built as described from G.
    return has_temporal_component(𝒢, 2|E| + 1 + k)
```

In conclusion, even basic questions like finding temporal components are computationally hard in temporal graphs. In this case, the problem turns out to be hard in both the strict and the non-strict settings. Interestingly, there exists problems that are hard only in the strict setting, and others only in the non-strict setting. Both settings are incomparable!

11.1 Temporal spanners

Given a (static) graph $G = (V, E)$, recall that a **spanning tree** of G is a subgraph $G' = (V', E') \subseteq G$ such that $V' = V$ and G' is a tree (see fig. 3). Clearly, spanning trees always

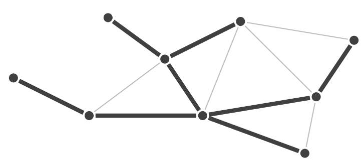


Figure 3: A spanning tree of a graph.

exist if G is connected, and we have already seen that computing them is an easy problem.

Given a temporal graph $\mathcal{G} = (V, E, \lambda)$ that is temporally connected, we can similarly define a **temporal spanning tree** of $\mathcal{G}$ as a subgraph $\mathcal{G}' = (V', E', \lambda') \subseteq \mathcal{G}$ such that $V' = V$, the footprint (V', E') is a tree, and $\mathcal{G}'$ is temporally connected. (See fig. 4.)

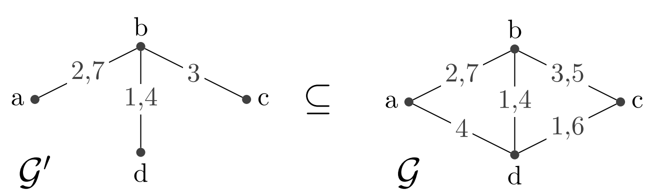


Figure 4: A temporal graph $\mathcal{G} \in$ TC (right) and one of its temporal spanning tree $\mathcal{G}'$ (left).

Is the existence of a temporal spanning tree guaranteed as long as the initial graph is in TC? Alas, no, for example the graph of fig. 5 does not admit a temporal spanning tree, although it is temporally connected.

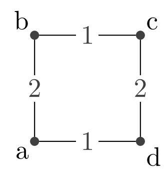


Figure 5: A temporal graph in TC that does not admit a temporal spanning tree.

In fact, even deciding if a given temporal graph admits a temporal spanning tree turns out to be NP-hard!

11.1.1 Relaxing trees into spanners

Since temporal spanning trees do not always exist, we need to replace this notion by a more flexible one. Given a temporal graph $\mathcal{G} = (V, E, \lambda)$ in TC, a **temporal spanner** of $\mathcal{G}$ is a subgraph $\mathcal{G}' = (V', E', \lambda') \subseteq \mathcal{G}$ such that $V' = V$ and $\mathcal{G}'$ is in TC.

Naturally, we want such a spanner to be as small as possible. Ideally, we would like that it uses only $O(n)$ edges. Is this always possible? Alas, no again! Hypercubes are an example of a class of graphs that have $\Theta(n \log n)$ edges. These edges can be labeled in a way that the graph is TC but none of the edges (or labels) can be removed. An example of hypercube in dimension 3 is given in fig. 6.

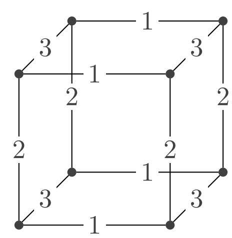


Figure 6: A temporal hypercube (of dimension 3) that is minimal in TC.

In fact, there exists even temporal graphs in TC with $\Theta(n^2)$ edges, none of which can be removed. Bad day for spanners!

11.1.2 Small spanners in temporal cliques

Given the above negative results, a natural question is whether small spanners are guaranteed at least in some restricted families of temporal graphs. The case of temporal cliques is an interesting one. A **temporal clique** is a temporal graph whose footprint is a complete graph. An example of temporal clique is given in fig. 7.

When the labeling of the clique is simple and proper (as in this example), we do have the guarantee that a spanner of size $O(n \log n)$ always exists whatever the labels. In fact,

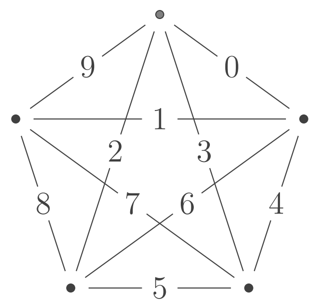


Figure 7: A temporal clique on 5 vertices.

most of these temporal cliques even admit spanners of size $2n-3$ that can be found easily using a technique called *dismountability*.

Let $\mathcal{G} = (V, E, \lambda)$ be a (simple and proper) temporal clique. For any node $v \in V$, denote by $e^-(v)$ the earliest edge incident to v, and by $n^-(v)$ the corresponding neighbor (its earliest neighbor). Define similarly the latest edge $e^+(v)$ and latest neighbor $n^+(v)$.

Because $\mathcal{G}$ is a clique, the fact that $n^-(v) = u$ for some u implies that u can reach every other node through v (indeed, v still has a later edge with every other node after time $\lambda(uv)$). Likewise, if $n^+(w) = u$, then every node can reach u through w. Let u, v, w be three nodes that satisfy $n^-(v) = u$ and $n^+(w) = u$. Then u can *delegate* its emissions to v, and delegate its receptions to w. These observations suggest a possible approach for self-reducing the problem of finding a temporal spanner as follows:

Theorem 11.3 (Dismountability)**.** *Let $\mathcal{G}$ be a temporal clique, and let u, v, w be three nodes such that $n^-(v) = u$ and $n^+(w) = u$. Let S' be a temporal spanner of $\mathcal{G} \setminus \{u\}$ Then $S := S' \cup \{uv, uw\}$ is a temporal spanner of $\mathcal{G}$.*

In this case, we say that node u is dismountable, and by extension, the clique is dismountable if at least one node is dismountable. A temporal clique is *recursively dismountable* if one can find an ordering of V that allows for a complete dismounting of the graph (down to $n = 2$), like in the example shown in fig. 8, together with the resulting spanner.

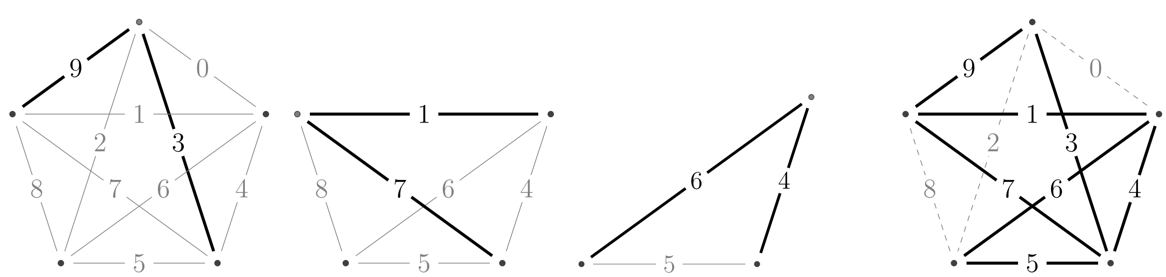


Figure 8: Recursive dismounting of a temporal clique and the resulting spanner (in bold).

Observe that exactly 2 edges are included in the spanner in every dismounting step. For a temporal clique with n vertices, this produces spanners of size $2n-3$. While most of the temporal cliques are dismountable, the recursion may fail at some point because not all of them are dismountable. In this case, the $O(n \log n)$ result is obtained by looking at the extra structure that non-dismountable cliques offer, and exploiting this structure using different techniques.

Do temporal cliques always admit $O(n)$-size spanners? As of today, we do not know. This is a major open question in the field.