Temporal graphs (Spring 2024)

3. Some facts about temporal graphs

Teacher: Arnaud Casteigts

In this third (and last) lesson, we discuss a number of features of temporal graphs, which differ from static graphs.

3.1 Time matters more than structure

Consider the temporal graph given in fig. 1 and suppose that its schedule repeats periodically (say, every 5 time units). Looking at the structure of the graph, i.e. its footprint only, node c is clearly the most central, and nodes a and e appear to be similarly excentered. However, the role of these nodes changes dramatically when time is added: node a can reach every other node within one full period, whereas node e may take up to 4 periods to reach node a, and node c has an intermediate score at this game.

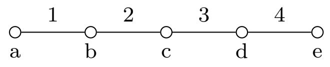


Figure 1: A temporal graph that repeats every 5 time units.

The conclusion of this simple observation is that classical centrality measures based on the structure of the graph only have very limited relevance. What matters is time!

3.2 Reachability is non-symmetric and non-transitive

In contrast to static graphs, the reachability relation in a temporal graph is *non-symmetric*: $u \rightsquigarrow v$ does not imply $v \rightsquigarrow u$. In this respect, temporal graphs are closer in spirit to directed graphs. However, unlike directed graphs, the reachability relation is also *non-transitive*: having $u \rightsquigarrow v$ and $v \rightsquigarrow w$ does not imply $u \rightsquigarrow w$, as illustrated in fig. 2.

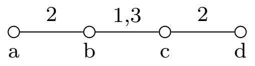


Figure 2: Reachability is non-transitive (e.g. $a \rightsquigarrow c$ and $c \rightsquigarrow d$, but $a \not\rightsquigarrow d$)

3.3 Connected components

A **connected component** in a temporal graph is a subset of vertices $V' \subseteq V$ such that for all $u, v \in V'$, we have $u \rightsquigarrow v$. An important effect of non-transitivity is that maximal connected components may overlap, as illustrated in fig. 3.

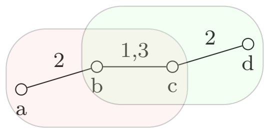


Figure 3: Overlapping components

This overlapping property is typical of temporal graphs. In static graphs (directed or not), maximal components do not overlap, they partition V into disjoint subsets. Unfortunately, non-transitivity is a source of computational hardness. This is easy to see in the case of connected components, as it allows us to design simple reductions from the classical NP-hard CLIQUE problem (in static graphs) to the CONNECTED COMPONENT problem in temporal graphs. In the case of the "strict setting" (only strict temporal paths are allowed), the reduction is straightforward.

Theorem 1. STRICT CONNECTED COMPONENT *is NP-hard.*

Proof. Given an instance of 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. 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, the connected components in $\mathcal{G}$ are exactly the cliques in G, and consequently:

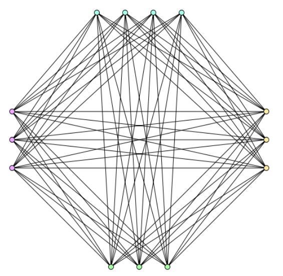
A clique of size k exists in G if and only if a connected component of size k exists in $\mathcal{G}$.

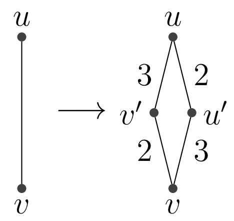
This shows that an algorithm for STRICT CONNECTED COMPONENT can be used to solve the CLIQUE problem (which is NP-hard). Thus, this problem is also NP-hard. □

Clearly, the above construction does not work in the non-strict setting, because the constructed temporal graph would then form a single component. A slightly more complicated reduction can be designed, that works for both the strict and the non-strict settings.

Theorem 2. NON-STRICT CONNECTED 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 new extra vertices, and the labels along the cycle are respectively set to $2, 3, 2, 3$ (see fig. 4). As before, we obtain that two vertices u and v in V


(a) An instance of CLIQUE.



(b) The semaphore gadget.

Figure 4: Reduction from CLIQUE to CONNECTED COMPONENT in temporal graphs.

can reach each other in $\mathcal{G}$ if and only if they share an edge in G. However, some of the extra vertices can also reach each other, which prevents us from concluding that components in $\mathcal{G}$ have the same size as cliques in G. To circumvent this problem, we add a new external vertex x and add an edge between x and every *extra* vertex, with labels 1 and 4. Note that all the $2 \cdot |E|$ extra vertices are now in the same component (through x). These vertices can also reach (and be reached by) all the original vertices. But this construction does not increase the reachability *among* original vertices. Consequently, we have:

A clique of size k exists in G iff a connected component of size $2 \cdot |E| + 1 + k$ exists in $\mathcal{G}$.

This shows that an algorithm for NON-STRICT CONNECTED COMPONENT can be used to solve the CLIQUE problem (which is NP-hard). Thus, this problem is also NP-hard. □

In this case, the problem turns out to be hard in both the strict and the non-strict settings. However, there exists problems that are hard only in the strict setting, and others that are hard only in the non-strict setting. Both settings are really incomparable!

3.4 Always-connected graphs and diameters

The **diameter** of a static graph G is the maximum distance between any pair of vertices. In temporal graphs, this notion can be extended in various ways. One of them is the **temporal diameter**, defined as the maximum time needed for any pair of vertices to reach each other. When seeing the temporal graph as a sequence of snapshots $\mathcal{G} = \langle G_1, G_2, G_3, ...\rangle$, is it possible that every snapshot has a small diameter and yet, the temporal diameter is large? (This question makes sense only in the strict setting.)

It turns out the answer is yes, this could be the case, and so, to a very large extent. To see this, it is convenient to pretend that the snapshots are chosen by an adversary that tries to slow down a temporal path from some node u to all the other vertices. Even if the diameter is only 3, the adversary can delay u from reaching some nodes until $n - 1$ time steps!

The temporal graph is built as follows. At each time steps, V is divided into two parts: one part contains all the vertices that have already been reached by u, and the other part contains the others. Each part is connected as a clique, and a single edge is added between the two parts (see fig. 5). Initially, u is alone in the informed part. Then, the newly informed

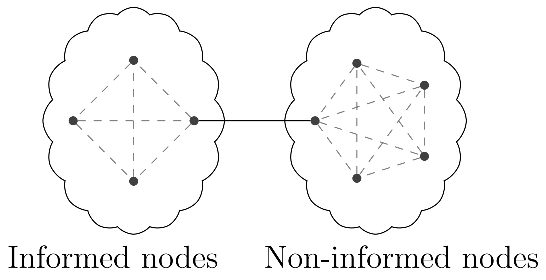


Figure 5: Small diameter vs. large temporal diameter.

vertex after G_1 joins the informed part in G_2, and so on. More generally, each G_i contains a clique of i informed vertices, connected by a single edge to a vertex in a clique of size $n - i$. Clearly, such a process will inform exactly one new vertex in each time step, and terminate only after $n - 1$ time steps.

3.5 Temporal spanners

Given a connected graph $G = (V, E)$, a **spanning tree** of G is a subgraph $G' = (V', E') \subseteq G$ such that $V' = V$ and G' is a tree (see fig. 6). Spanning trees always exist. Furthermore,

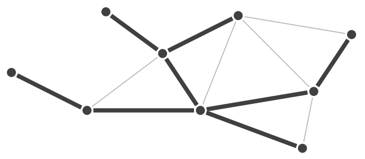


Figure 6: A spanning tree of a graph.

finding a spanning tree 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. 7.)

Is the existence of a temporal spanning tree guaranteed as long as the initial graph is temporally connected? In the case of fig. 7, it does. But in general, it does not, and deciding if a given temporal graph admits a temporal spanning tree is (again) an NP-hard problem!

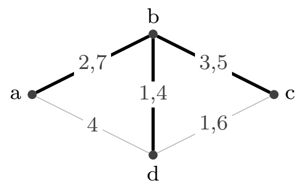


Figure 7: A temporal spanning tree.

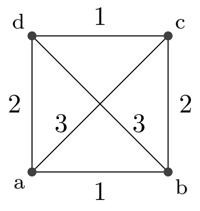
3.6 Exercises

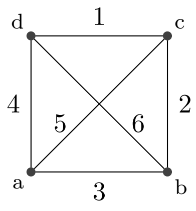
1. Could we have the opposite effect as the one discussed in section 3.4, namely, having a large diameter in each snapshot (say, diameter $n-1$ in each G_i), and yet allow a specific vertex to reach all the others fast? We can indeed! Find how, and how quick this could be done (you are designing the snapshots here). **(Solved in class.)**

2. Find a temporal graph that is temporally connected but does not admit a temporal spanning tree. **(Solved in class.)**

3. Since temporal spanning trees do not always exist, we want to replace this notion by a more flexible one: given a temporal graph $\mathcal{G}$, we want to remove as many edges as possible from its footprint, while preserving temporal connectivity. The remaining graph is called a **temporal spanner** of $\mathcal{G}$.

   For each of the following graphs, how many edges does a minimum size spanner have? Indicate the corresponding set of edges to be removed (simultaneously):


d
c
a
b
1
2
3
3
2
1


d
c
a
b
1
4
5
6
2
3