Forbidden Patterns in Temporal Graphs Resulting from Encounters in a Corridor

Michel Habib[1], Minh-Hang Nguyen[1], Mikaël Rabie[1], and Laurent Viennot[2]

[1] Université Paris Cité, CNRS, IRIF, F-75013, Paris, France
[2] Université Paris Cité, CNRS, Inria, IRIF, F-75013, Paris, France
firstName.lastName@irif.fr

Abstract. In this paper, we study temporal graphs arising from mobility models where some agents move in a space and where edges appear each time two agents meet. We propose a rather natural one-dimensional model.
If each pair of agents meets exactly once, we get a temporal clique where each possible edge appears exactly once. By ordering the edges according to meeting times, we get a subset of the temporal cliques. We introduce the first notion of of forbidden patterns in temporal graphs, which leads to a characterization of this class of graphs. We provide, thanks to classical combinatorial results, the number of such cliques for a given number of agents.
We consider specific cases where some of the nodes are frozen, and again provide a characterization by forbidden patterns. We give a forbidden pattern when we allow multiple crossings between agents, and leave open the question of a characterization in this situation.

Keywords: Temporal graphs, mobility models, forbidden patterns, mobile clique

1 Introduction

1.1 Motivation

Temporal graphs arise when the edges of a graph appear at particular points in time (see e.g. [4,11,14]). Many practical graphs are indeed temporal from social contacts, co-authorship graphs, to transit networks. A very natural range of models for temporal graphs comes from mobility. When agents move around a space, we can track the moments when they meet each other and obtain a temporal graph. We ask how to characterize temporal graphs resulting from such a mobility model.

A classical model used for mobile networks is the unit disk graph where a set of unit disks lie in the plane, and two disk are linked when they intersect. When the disks are moving, we obtain a so-called dynamic unit disk graph [19], and the appearance of links then forms a temporal graph. We consider a one-dimensional version where the disks are moving along a line or equivalently a narrow corridor of unit width. This could encompass practical settings such as communicating cars on a single road. We further restrict to the sparse regime where each disk intersects at most one other disk at a time. In other words, the edges appearing at any given time always form a matching. This restriction, called local injectivity, has already been considered in the study of temporal cliques [5] which are temporal graphs where an edge between any pair of nodes appears exactly once.

When two agents can communicate when they meet, one can ask how information can flow in the network. The appropriate notion of connectivity then arises from temporal paths which are paths where edges appear one after another along the path. A temporal spanner can then be defined as a temporal subgraph that preserves connectivity. An interesting question concerning temporal graphs is to understand which classes of temporal graphs have temporal spanners of linear size. Although some temporal graphs have only $\Theta(n^2)$-size temporal spanners [12], temporal cliques happen to have $O(n \log n)$-size temporal spanners [5]. A natural question is whether temporal graphs resulting from a mobility model can have sparse spanners. In particular, do temporal cliques arising from our 1D model have temporal spanners of linear size?

1.2 Our Contribution

Our main contribution is a characterization of the temporal cliques that result from this 1D model. A temporal clique can only arise when agents start out in a certain order along the corridor and end up in the opposite order after crossing each other exactly once. We provide a characterization of such temporal cliques in terms of forbidden ordered patterns on three nodes. This characterization leads directly to an $O(n^3)$-time algorithm for testing whether an ordering of the n nodes of a temporal clique is appropriate and allows to exclude these patterns. Interestingly, an $O(n^2)$-time algorithm allows to find such an appropriate initial ordering of the nodes from the list of the edges of the clique ordered by appearance time. Moreover, we can actually check in $O(n^2)$ time that this order excludes the forbidden patterns to obtain an overall linear-time recognition algorithm, since we have $n(n-1)/2$ edges in our graphs.

Another way of looking at this problem is sorting through adjacent transpositions an array A, where n elements are initially stored in reverse order. At each step, we choose an index i such that $A[i] > A[i+1]$ and swap the two elements at positions i and $i+1$. The array is guaranteed to be sorted in increasing order after $T = n(n-1)/2$ steps, since the permutation of the elements in A has initially T inversions while each step decreases this number by one. Note that this is reminiscent of bubble sorting, which indeed operates according to a sequence of such transpositions. This naturally connects our 1D model to the notion of reduced decompositions of a permutation [17]. A classical combinatorial result gives a formula for the number of temporal cliques with n nodes resulting from our 1D model.

As far as we know, we introduce the first definition of forbidden patterns in a temporal graph. Our definition is based on the existence of an order on the nodes (which actually corresponds to their initial order along the line). A forbidden pattern is a temporal subgraph with a relative ordering of its nodes, and with a forbidden relative ordering of its edges according to their time labels.

In addition, we show that our temporal cliques do contain temporal spanners of linear size (with exactly $2n-3$ edges) by enlightening a convenient temporal subgraph that considers only the edges having, as one of their endpoints, one of the two extreme agents in the initial order along the line.

Finally, we consider some generalizations. First, what happens if some agents do not move. In particular, we are no longer working on a clique, since edges between two frozen agents never occur. Second, we consider at what might be a forbidden pattern definition if edges can occur multiple times, that is when some pairs of agents can cross each other multiple times.

1.3 Related Works

Dynamic unit disk graph. A closely related work concerns the detection of dynamic unit disk graphs on a line [18,19]. An algorithm is proposed to decide whether a continuous temporal graph can be embedded in the line along its evolution, such that the edges present at each time instant correspond to the unit disk graph within the nodes according to their current position in the embedding at that time. The sequence of edge events (appearance or disappearance) is processed online one after another, relying on a PQ-tree to represent all possible embeddings at the time of the current event according to all events seen so far. It runs within a logarithmic factor from linear time. Our model is tailored for discrete time and assumes that two nodes cross each other when an edge appears between them. This is not the case in the dynamic unit disk graph model: an edge can appear during a certain period of time between two nodes even if they don't cross each other. The PQ-tree approach can probably be adapted to our model for a more general recognition of the temporal graphs it produces. Note that our characterization leads to a faster linear-time algorithm for recognizing temporal cliques arising from our model.

Temporal Graph. Temporal graphs (also known as dynamic, evolving or time-varying networks) can be informally described as graphs that change with time, and are an important topic in both theory and practice when there are many of real-world systems that can be modelled as temporal graphs, see [11]. The problem of temporal connectivity has been considered, by Awerbuch and Even [1], and studied more systematically in [12].

Forbidden Patterns. Since the seminal papers [6,15], many hereditary graph classes have been characterized by the existence of an order of the vertices that avoids some pattern, i.e. an ordered structure. These include bipartite graphs, interval graphs, chordal graphs, comparability graphs and many others. In [10], it is proved that any class defined by a set of forbidden patterns on three nodes can be recognized in $O(n^3)$ time. This was later improved in [7] with a full characterization of the 22 graph classes that can be defined with forbidden patterns on three nodes. An interesting extension to forbidden circular structures is given in [9]. The growing interest in forbidden patterns in the study of hereditary graph classes is partly supported by the certification that such an ordering avoiding the patterns provides for a recognition algorithm in the YES case.

Reduced decomposition. The number of reduced decompositions of a permutation of n elements is studied in [16]. An explicit formula is given for the reverse permutation $n, n-1, \ldots, 1$ based on the hook length formula [2,8].

1.4 Roadmap

In Section 2, we introduce the notions that we will use throughout the paper. In particular, we provide the definitions of temporal graphs and 1D mobility models. Section 3 provides our main results: a characterization of mobility cliques through forbidden patterns, the number of cliques of a given size, a detection algorithm, and a linear size spanner of the graph. Sections 4 and 5 give results on two extensions of the model. The first one considers the case where some pairs of agents do not cross, the second one provides a forbidden pattern in the case where pairs can cross each other several times. Finally, we raise some open questions and perspectives in Section 6.

2 Preliminaries and Mobility Model

In this section, we introduce the definitions and notations we will use through the paper. In particular, we first define formally temporal graphs and forbidden patterns. We then introduce the mobility model and related combinatoric concepts.

2.1 Temporal Graphs and Forbidden Patterns

Informally, a temporal graph is a graph with a fixed vertex set and whose edges change with time. A *temporal graph* can be formally defined as a pair $\mathcal{G} = (G, \lambda)$ where $G = (V, E)$ is a graph with vertex set V and edge set E, and $\lambda : E \to 2^{\mathbb{N}}$ is a labeling assigning to each edge $e \in E$ a non-empty set $\lambda(e)$ of discrete times when it appears. We note $uv \in E$ the edge between the pair of vertices (or nodes) u and v. If λ is locally injective in the sense that adjacent edges have disjoint sets of labels, then the temporal graph is said to be *locally injective*. If λ is additionally *single valued* (i.e. $|\lambda(e)| = 1$ for all $e \in E$), then (G, λ) is said to be simple [5]. The maximum time label $T = \max \cup_{e \in E} \lambda(e)$ of an edge is called the *lifetime* of (G, λ). In the sequel, we will mostly restrict ourselves to simple temporal graphs and even require the following restriction of locally injective. A temporal graph is *incremental* if at most one edge appears in each time step, that is $\lambda(e) \cap \lambda(f) = \emptyset$ for any distinct $e, f \in E$.

A (strict) *temporal path* is a sequence of triplets $(u_i, u_{i+1}, t_i)_{i \in [k]}$ such that $(u_1, \ldots, u_{k+1})$ is a path in G with increasing time labels where its edges appear: formally, for all $i \in [k]$, we have $u_i u_{i+1} \in E$, $t_i \in \lambda(u_i u_{i+1})$ and $t_i < t_{i+1}$. Note that our definition corresponds to the classical strict version of temporal path as we require time labels to strictly increase along the path[3]. A temporal graph is *temporally connected*, if every vertex can reach any other vertex through a temporal path. A *temporal sub-graph* (G', λ') of a temporal graph (G, λ) is a temporal graph such that G' is a subgraph of G and λ' satisfies $\lambda'(e) \subseteq \lambda(e)$ for all $e \in E'$. A *temporal spanner* of $\mathcal{G}$ is a temporal sub-graph $\mathcal{H}$ preserving temporal connectivity, that is there exists a temporal path from u to v in $\mathcal{H}$ whenever there exists one in $\mathcal{G}$.

A representation $\mathcal{R}$ of a temporal graph $\mathcal{G} = ((V, E), \lambda)$ is defined as an ordered list of $M = |\lambda| = \sum_{e \in E} |\lambda(e)|$ triplets $\mathcal{R} = (u_1, v_1, t_1), \ldots, (u_M, v_M, t_M)$ where each triplet (u_i, v_i, t_i) indicates that edge $u_i v_i$ appears at time t_i. We additionally require that the list is sorted by non-decreasing time. In other words, we have $\lambda(uv) = \{t_i : \exists i \in [M], (u_, v_i, t_i) \in \mathcal{R}\}$ for all $uv \in E$. Note that any incremental temporal graph $\mathcal{G}$ has a unique representation denoted by $\mathcal{R}(\mathcal{G})$. Indeed, its temporal connectivity only depends on the ordering in which edges appear, we can thus assume without loss of generality that we have $\cup_{e \in E} \lambda(e) = [T]$ where T is the lifetime of $((V, E), \lambda)$ (we use the notation $[T] = \{1, \ldots, T\}$). Given two incremental temporal graphs $\mathcal{G} = ((V, E), \lambda)$ and $\mathcal{G}' = ((V', E'), \lambda')$, an *isomorphism* from $\mathcal{G}$ to $\mathcal{G}'$ is a one-to-one mapping $\phi : V \to V'$ such that, for any $u, v \in V$, $uv \in E \Leftrightarrow \phi(u)\phi(v) \in E'$ (ϕ is a graph isomorphism), and their representation $\mathcal{R}(\mathcal{G}) = (u_1, v_1, t_1), \ldots, (u_M, v_M, t_M)$ and $\mathcal{R}(\mathcal{G}') = (u'_1, v'_1, t'_1), \ldots, (u'_M, v'_M, t'_M)$ have same length $M = |\lambda| = |\lambda'|$ and are temporally equivalent in the sense that edges appear in the same order: $u_i v_i = u'_i v'_i$ for all $i \in [M]$. When such an isomorphism exists, we say that $\mathcal{G}$ and $\mathcal{G}'$ are *isomorphic*.

A *temporal clique* is a temporal graph (G, λ) where the set of edges is complete, and where we additionally require the temporal graph to be incremental and λ to be single valued. Notice that it is a slight restriction compared to the definition of [5] which requires (G, λ) to be locally injective rather than incremental. However, we do not lose in generality as one can easily transform any locally injective temporal graph into an incremental temporal graph with same temporal connectivity (we simply stretch time by multiplying all time labels by n^2 and arbitrarily order edges with same original time label within the corresponding interval of n^2 time slots in the stretched version). With a slight abuse of notation, we then denote the label of an edge uv by $\lambda(uv) \in \mathbb{N}$.

A *temporal pattern* is defined as an incremental temporal graph $\mathcal{H} = (H, \lambda)$. An incremental temporal graph $\mathcal{G} = (G, \lambda')$ *excludes* $\mathcal{H}$ when it does not have any temporal sub-graph $\mathcal{H}'$ which is isomorphic to $\mathcal{H}$. A temporal pattern *with forbidden edges* is a temporal pattern $\mathcal{H} = (H, \lambda)$ together with a set $F \subseteq V \times V \setminus E$ of forbidden edges in $H = (V, E)$. An incremental temporal graph

[3] The interested reader can check that the two notions of strict temporal path and non-strict temporal path are the same in locally injective temporal graphs.

$\mathcal{G} = ((V', E'), \lambda')$ *excludes* $\mathcal{H}$ when it does have any temporal sub-graph $\mathcal{H}'$ which is isomorphic to (H, λ') through an isomorphism ϕ respecting non-edges, that is any pair of nodes $u, v \in V'$ which is mapped to a forbidden edge $\phi(u)\phi(v) \in F$, we have $uv \notin E'$.

An *ordered temporal graph* is a pair $(\mathcal{G}, \pi)$, where $\mathcal{G}$ is a temporal graph and π is an ordering of its nodes. Similarly, an *ordered temporal pattern* $(\mathcal{H}, \pi)$ is a temporal pattern $\mathcal{H}$ together with an ordering π of its node. An ordered incremental temporal graph $(\mathcal{G}, \pi')$ *excludes* $(\mathcal{H}, \pi)$ when it does not have any temporal sub-graph $\mathcal{H}'$ which is isomorphic to $\mathcal{H}$ through an isomorphism ϕ preserving relative orderings, that is $\pi(\phi(u)) < \pi(\phi(v))$ whenever $\pi'(u) < \pi'(v)$. We then also say that the ordering π' *excludes* $(\mathcal{H}, \pi)$ *from* $\mathcal{G}$, or simply excludes $(\mathcal{H}, \pi)$ when $\mathcal{G}$ is clear from the context. We also define an ordered temporal pattern with forbidden edges similarly as above.

2.2 1D-Mobility Model

We introduce here the notion of temporal graph associated to mobile agents moving along a line that is a one-dimensional space. Consider n mobile agents in an oriented horizontal line. At time $t_0 = 0$, they initially appear along the line according to an ordering π_0. These agents move in the line and can cross one another as time goes on. We assume that a crossing is always between exactly two neighboring agents, and a single pair of agents cross each other at a single time. By ordering the crossings, we have the kth crossing happening at time $t_k = k$.

A *1D-mobility schedule* from an ordering $\pi_0 = a_1, \ldots, a_n$ of n agents is a sequence $x = x_1, \ldots, x_T$ of crossings within the agents. Each crossing x_t is a pair uv indicating that agents u and v cross each other at time t. Note that their ordering π_t at time t is obtained from π_{t-1} by exchanging u and v, and it is thus required that they appear consecutively in π_{t-1}. To such a schedule, we can associate a temporal graph $\mathcal{G}_{\pi_0,x} = ((V, E), \lambda)$ such that:

- $V = \{a_1, \ldots, a_n\}$,
- $E = \{uv : \exists t \in [T], x_t = uv\}$,
- for all $uv \in E$, $\lambda(uv) = \{t : x_t = uv\}$.

We are interested in particular by the case where all agents cross each other exactly once as the resulting temporal graph is then a temporal clique which is called *1D-mobility temporal clique.* More generally, we say that an incremental temporal graph $\mathcal{G}$ *corresponds to a 1D-mobility schedule* if there exists some ordering π of its vertices and a 1D-mobility schedule x from π such that the identity is an isomorphism from $\mathcal{G}$ to $\mathcal{G}_{\pi,x}$. It is then called a *1D-mobility temporal graph.*

2.3 Reduced Decomposition of a Permutation

Our definition of mobility model is tightly related to the notion of reduced decomposition of a permutation [17]. Let $\mathcal{S}_n$ denote the symmetric group on n elements. We represent a permutation $w \in S_n$ as a sequence $w = w(1), \ldots, w(n)$ and define its length $l(w)$ as the number of inverse pairs in w, i.e. $l(w) = |\{i, j : i < j, w(i) > w(j)\}|$. A *sub-sequence* w' of w is defined by its length $k \in [n]$ and indices $1 \leq i_1 < \cdots < i_k \leq n$ such that $w' = w(i_1), \ldots, w(i_k)$.

A *transposition* $\tau = (i, j)$ is the transposition of i and j, that is $\tau(i) = j$, $\tau(j) = i$ and $\tau(k) = k$ for $k \in [n] \setminus \{i, j\}$. It is an *adjacent* transposition when $j = i + 1$. Given a permutation w and an adjacent transposition $\tau = (i, i + 1)$, we define the *right product* of w by τ as the composition $w\tau = w \circ \tau$. Note that $w' = w\tau$, as a sequence, is obtained from w by exchanging the numbers in positions i and $i + 1$ as we have $w'(i) = w(\tau(i)) = w(i + 1)$, $w'(i + 1) = w(\tau(i + 1)) = w(i)$ and

$w'(k) = w(k)$ for $k \neq i, j$. A *reduced decomposition* of a permutation $w \in S_n$ with length $l(w) = l$, is a sequence of adjacent transpositions $\tau_1, \tau_2, \ldots, \tau_l$ such that we have $w = \tau_1 \ldots \tau_l$. Counting the number of reduced decompositions of a permutation has been well studied (see in particular [16]).

The link with our 1D-mobility model is the following. Consider a 1D-mobility schedule x from an ordering π_0. Without loss of generality we assume that agents are numbered from 1 to n. Each ordering π_t is then a permutation. If agents u and v cross at time t, i.e. $x_t = uv$, and their positions in π_{t-1} are i and $i+1$, we then have $\pi_t = \pi_{t-1}\tau_t$ where $\tau_t = (i, i+1)$. If each pair of agents cross at most once, then one can easily see that the schedule x of crossings corresponds to a reduced decomposition $\tau_1, \ldots, \tau_T$ of $\pi_0^{-1}\pi_T = \tau_1 \cdots \tau_T$ as the ending permutation is $\pi_T = \pi_0\tau_1 \cdots \tau_T$. Note that this does not hold if two agents can cross each other more than once as the length of the schedule can then be longer than the length of $\pi_0^{-1}\pi_T$.

Interestingly, another decomposition is obtained by interpreting the crossing $x_t = uv$ at time t as the transposition (u, v). We then have $\pi_t = x_t\pi_{t-1}$ for each time t, and finally obtain $\pi_T = x_T \cdots x_1\pi_0$. Note that given an arbitrary sequence of transpositions $x_1, \ldots, x_T$, it is not clear how to decide whether there exists an ordering π_0 and a corresponding sequence of *adjacent* transpositions $\tau_1, \ldots, \tau_T$ such that $x_t \cdots x_1\pi_0 = \pi_0\tau_1 \cdots \tau_t$ for all $t \in [T]$. This is basically the problem we address in the next section.

3 1D-Mobility Temporal Cliques

3.1 Characterization

Consider the ordered temporal patterns from Figure 1 with respect to the initial ordering of the nodes in a 1D-mobility schedule x producing a temporal clique $\mathcal{G}_x$. One can easily see that the upper-left pattern cannot occur in $\mathcal{G}_x$ within three agents a, b, c appearing in that order initially: a and c cannot cross each other as long as b is still in-between them, while the pattern requires that edge ac appears before ab and bc. A similar reasoning prevents the presence of the three other patterns. It appears that excluding these four patterns suffices to characterize 1D-mobility temporal cliques, as stated bellow.

Theorem 1. *A temporal clique is a 1D-mobility temporal clique if and only if there exists an ordering of its nodes that excludes the four ordered temporal patterns of Figure 1.*

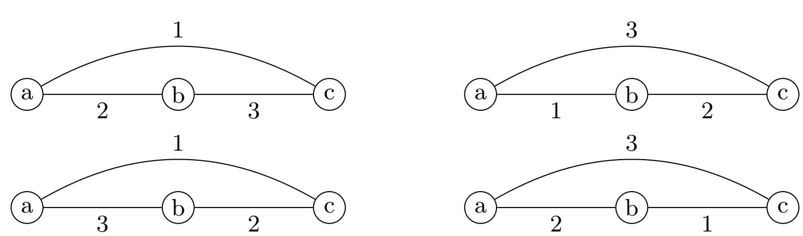

Fig. 1. Ordered forbidden patterns in an ordered 1D-mobility temporal clique. Each pattern is ordered from left to right and has associated ordering a, b, c.

Let $\mathcal{C}$ denote the class of temporal cliques which have an ordering excluding the four ordered temporal patterns of Figure 1. We first prove that any 1D-mobility temporal clique is in $\mathcal{C}$:

Proposition 1. *For any 1D-mobility schedule x from an ordering π of n agents such that $\mathcal{G}_{\pi,x} = ((V,E),\lambda)$ is a temporal clique, the initial ordering π excludes the four patterns of Figure 1.*

This proposition is a direct consequence of the following lemma.

Lemma 1. *Consider three nodes $a, b, c \in V$ such that time $\lambda(ac)$ happens in-between $\lambda(ab)$ and $\lambda(bc)$, i.e. $\lambda(ac)$ is the median of $\{\lambda(ab), \lambda(ac), \lambda(bc)\}$, then b is in-between a and c in the initial ordering, i.e. either a, b, c or c, b, a is a sub-sequence of π.*

Proof. For the sake of contradiction, suppose that b is not in-between a and c initially. At time $\min\{\lambda(ab), \lambda(ac)\}$, it first crosses a or c, and it is now in-between a and c. As a and c cannot cross each other as long as b lies in-between them, the other crossing with of b with a or c should thus occur before $\lambda(ac)$, implying $\max\{\lambda(ab), \lambda(ac)\} < \lambda(ac)$, in contradiction with the hypothesis. The only possible initial orderings of these three nodes are thus a, b, c and c, b, a.

One can easily check that the above Lemma forbids the four patterns of Figure 1. Indeed, in each pattern, the edge of label 2 that appears in-between the two others in time, is adjacent to the middle node while it should link the leftmost and rightmost nodes. Proposition 1 thus follows.

We now show that forbidding these four patterns fully characterizes 1D-mobility temporal cliques. For that purpose, we construct a mapping from ordered temporal cliques in $\mathcal{C}$ to the set $R(w_n)$ of all reduced decompositions of w_n where $w_n = n, n-1, \ldots, 1$ is the permutation in $\mathcal{S}_n$ with longest length.

Lemma 2. *Any temporal clique $\mathcal{G} \in \mathcal{C}$ having an ordering π excluding the four patterns of Figure 1, can be associated to a reduced decomposition $f(\mathcal{G}, \pi)$ of w_n. Moreover, the representation $\mathcal{R}(\mathcal{G})$ of $\mathcal{G}$ corresponds to a 1D-mobility schedule starting from π and $\mathcal{G}$ is a 1D-mobility temporal clique.*

Proof. Recall that, up to isomorphism, we can assume that $\mathcal{G}$ has lifetime $T = n(n-1)/2$ and that exactly one edge appears at each time $t \in [T]$. Consider the corresponding representation $\mathcal{R}(\mathcal{G}) = (u_1, v_1, 1), (u_2, v_2, 2), \ldots, (u_T, v_T, T)$. Starting from the initial ordering $\pi_0 = \pi$, we construct a sequence $\pi_1, \ldots, \pi_T$ of orderings corresponding to what we believe to be the positions of the agents at each time step if we read the edges in $\mathcal{R}(\mathcal{G})$ as a 1D-mobility schedule. More precisely, for each $t \in T$, π_t is defined from π_{t-1} as follows. As the edge $u_t v_t$ should correspond to a crossing $x_t = u_t v_t$, it can be seen as the transposition exchanging u_t and v_t so that we define $\pi_t = x_t \pi_{t-1}$. Equivalently, we set $\tau_t = (i, j)$ where i and j respectively denote the indexes of u_t and v_t in π_{t-1}, i.e. $\pi_{t-1}(i) = u_t$ and $\pi_{t-1}(j) = v_t$. We then also have $\pi_t = \pi_{t-1}\tau_t$.

Our main goal is to prove that $f(\mathcal{G}, \pi) := \tau_1, \ldots, \tau_T$ is the desired reduced decomposition of w_n. For that, we need to prove that u_t and v_t are indeed adjacent in $\pi_{t-1} = \pi_0 \tau_1 \cdots \tau_{t-1} = x_{t-1} \cdots x_1 \pi_0$. For the sake of contradiction, consider the first time t when this fails to be. That is $\tau_1, \ldots, \tau_{t-1}$ are indeed adjacent transpositions, edge uv appears at time t, i.e. $uv = u_t v_t$, and u, v are not consecutive in π_{t-1}. We assume without loss of generality that u is before v in π_0, i.e. u, v is a sub-sequence of π_0. We will mainly rely on the following observation:

Consider two nodes x, y such that x is before y in π_0, then x is before y in π_{t-1} if and only edge xy appears at t or later, i.e. $\lambda(xy) \geq t$.

The reason comes from the assumption that $\tau_1, \ldots, \tau_{t-1}$ are all adjacent transpositions: as long as only x or y is involved in such a transposition, their relative order cannot change. The above observation thus implies in particular that u is still before v in π_{t-1}. Now, as u and v are not consecutive in π_{t-1}, there must exist an element w between elements u and v in π_{t-1}. We consider the two following cases:

Case 1. w was already in-between u and v in π_0, that is u, w, v is a sub-sequence of π_0. As the relative order has not changed between these three nodes, we have $\lambda(uw) > t$ and $\lambda(wv) > t$ as their appearing time is distinct from $t = \lambda(uv)$. This is in contradiction with the exclusion of the two patterns on the left of Figure 1.

Case 2. w was not in-between u and v in π_0. Consider the case where u, v, w is a sub-sequence of π_0. From the observation, we we deduce that $\lambda(vw) < t$ and $\lambda(uw) > t$, which contradicts with the exclusion of the bottom-right pattern of Figure 1. The other case where w, u, v is a sub-sequence of π_0 is symmetrical and similarly leads to a contradiction with the exclusion of the top-right pattern of Figure 1.

We get a contradiction in all cases and conclude that $\tau_1, \ldots, \tau_T$ are all adjacent transpositions. This implies that x is indeed a valid 1D-mobility schedule from π. As x is defined according to the ordering of edges in $\mathcal{R}(\mathcal{G})$ by appearing time, $\mathcal{G}$ is obviously isomorphic to $\mathcal{G}_{\pi,x}$.

Additionally, as each pair of elements occurs exactly in one transposition, the permutation $\tau_1 \cdots \tau_T$ has length $T = n(n-1)/2$ and must equal w_n. The decomposition $f(\mathcal{G}, \pi) = \tau_1, \ldots, \tau_T$ is thus indeed a reduced decomposition of w_n.

Theorem 1 is a direct consequence of Proposition 1 and Lemma 2.

3.2 Recognition Algorithm

We provide an algorithm that decides if a clique belongs to $\mathcal{C}$, and provides an ordering of the nodes that avoids the patterns if it is the case.

The main idea of the algorithm relies on Lemma 1 which allows to detect within a triangle which node should be in-between the two others in any ordering avoiding the patterns by checking the three times at which the edges of the triangle appear.

We assume to be given the input as a representation of the temporal graph, i.e. the list $\mathcal{R}(\mathcal{G})$ of the edges in the form (u, v, t), sorted according to their time labels. The algorithm runs in $O(n^2)$ time.

First we try to compute an ordering of the vertices with the function VertexSorting($\mathcal{R}(\mathcal{G})$). To do that, the subroutine ExtremalNodes(V) provides the two nodes that should be at the extremities of some subset V of nodes. It outputs these two nodes by excluding repeatedly a node out of some triplets again and again until only two nodes are left, using Lemma 1 to identify which one is in the middle.

We deduce the two extremities a and z of V, keep a as the first element. We then repeat $n-2$ times: add back z to the remaining nodes, compute the extremities. If z is one of the extremities, remove the other element and add it to the ordering. Otherwise, return $\perp$ as a contradiction has been found (z must always be an extremity if we have a 1D-mobility temporal clique).

We then need to check that each edge indeed switches two consecutive nodes one after another in the 1D-mobility model. To do that, we represent the sequence of permutations starting from π the initial ordering, and check that each switch, according to the edges sorted by time label, corresponds to an exchange between two consecutive nodes. If at some point, we try to switch non consecutive elements, we return $False$, otherwise at the end we proved that we had a 1D-mobility temporal clique and return $True$.

```
Function VertexSorting(R(G))
    Input: The representation R(G) of a temporal clique G = ((V, E), λ).
    Output: A vertex ordering π.
    Compute a matrix representing λ and the set V of vertices from R(G).
    X := ExtremalNodes(V)
    Let a and z be the two vertices in X.
    Define an ordering π with first element a.
    V := V \ {a}
    While V ≠ {z} do
        X := ExtremalNodes(V)
        If z ∉ X then
            return ⊥  /* Failure.                                  */
        else
            Let b be the node in X \ {z}.
            Append b to π.
            V := V \ {b}
    Append z to π.
    Return π

Function ExtremalNodes(V)
    Let W be a copy of V.
    Pick any pair u, v of nodes in W.
    W := W \ {u, v}
    Set X := {u, v}. /* Tentative pair of extremal nodes.          */
    While W ≠ ∅ do
        Remove a node w from W.
        X := TriangleExtremities(X ∪ {w})
    Return X

Function TriangleExtremities(T)
    Let u, v, w be the three nodes in T.
    Retrieve the three time labels λ(uv), λ(vw), λ(uw).
    Return the edge e ∈ {uv, vw, uw} with median time label.

Input: A temporal clique G = ((V, E), λ) given by its representation R(G).
Output: True if G excludes the four forbidden patterns of Figure 1, False otherwise.
π := VertexSorting(R(G))
If π = ⊥ then return False
Compute the index σ of each vertex.
For each triplet (u, v, t) in R(G) do
    If |σ(u) − σ(v)| = 1 then
        /* u and v are consecutive in π.                             */
        Swap u and v in π and update σ.
    else
        Return False
Return True
```

Algorithm 1: Vertex sorting and Recognition.

3.3 Counting

We now estimate the number $|\mathcal{C}|$ of 1D-mobility temporal cliques with n nodes through the following result.

Proposition 2. *The number of 1D-mobility temporal cliques with n nodes is*

$$|\mathcal{C}| = \frac{|R(w_n)|}{2} = \frac{1}{2} \frac{\binom{n}{2}!}{1^{n-1}3^{n-2} \cdots (2n-3)^1}.$$

Let us define $\mathcal{C}' \subseteq \mathcal{C} \times \mathcal{S}_n$ as the set of ordered temporal cliques $(\mathcal{G}, \pi)$ such that $\mathcal{G} = ((V, E), \lambda) \in \mathcal{C}$ and π is an ordering of V such that $\mathcal{R}(\mathcal{G})$ provides a 1D-mobility schedule from π. Proposition 2 derives from two following lemmas and known results [16] counting the number $|R(w_n)|$ of reduced decompositions of w_n according to the hook length formula [8].

Lemma 3. *The mapping $f : \mathcal{C}' \to R(w_n)$ defined in Lemma 2 is a bijection.*

Proof. We simply define a mapping $g : R(w_n) \to \mathcal{C}'$ such that $f \circ g$ is the identity. Consider a reduced decomposition $\rho = \tau_1, \ldots, \tau_T$ of w_n where each τ_t is an adjacent transposition. As w_n has $n(n-1)/2$ inversions, its length is indeed $T = n(n-1)/2$. Let $\pi_0 = 1, \ldots, n$ be the identity permutation and define $\pi_t = \pi_0 \tau_1 \cdots \tau_t = \tau_1 \cdots \tau_t$ for each $t \in [T]$. Let $x_t = uv$ be the pair of elements in position i and $i+1$ in π_{t-1} where i is the index such that $\tau_t = (i, i+1)$. We then have $\pi_t = x_t \pi_{t-1}$ for all $t \in [T]$, and x is indeed a 1D-mobility schedule from π_0 that leads to w_n. As each pair of agents $u, v \in \pi_0$ with $u < v$ appears as sub-sequence u, v in π_0 and sub-sequence v, u in $\pi_T = w_n$, they must cross at some time t such that $x_t = uv$. As the total number of crossings is $T = n(n-1)/2$, this can happen only once, and $\mathcal{G}_{\pi_0,x}$ is a temporal clique. We can thus define $g(\rho) = (\mathcal{G}_{\pi_0,x}, \pi_0)$ which satisfies $f(g(\rho)) = \rho$ as we have $x_t \cdots x_1 \pi_0 = \pi_t = \pi_0 \tau_1 \cdots \tau_t$ for all $t \in [T]$.

Lemma 4. *Any 1D-mobility temporal clique admits exactly two orderings π such that $\mathcal{R}(\mathcal{G})$ provides a 1D-mobility schedule from π.*

Proof. Assume that $\mathcal{G} = ((V, E), \lambda)$ is a 1D-mobility temporal clique. Let π' be an ordering that excludes the four forbidden patterns from $\mathcal{G}$.

Note that, by lemma 1, for any three nodes, u, v, w, of V, with edge label $\lambda(uw) < \lambda(uv) < \lambda(wv)$, then node w has to be in-between of nodes u and v in all orderings that excludes the forbidden patterns from $\mathcal{G}$.

For any $W \subseteq V$, the extremal vertices of W returned by ExtremalNodes(W) are uniquely defined by W. In other words, these extremal vertices do not depend on the order of nodes w picked from W in line 23, and two nodes, u and v, picked from W in line 19. The reason is that, in the loop from line 22 to line 24, all nodes that are in-between of some other two vertices in W, corresponding to the ordering π', will be eliminated one by one. Thus, function ExtremalNodes(W) return two extremal vertices among vertices in W corresponding to ordering π'.

We prove that the ordering π returned by function VertexSorting($\mathcal{R}(\mathcal{G})$) is either π' itself or the reverse of π'. Furthermore, the ordering π is uniquely defined, depending on the choice of the first element a in line 5.

In line 5, there are two possible choices to pick a node a in set X. At the beginning of the i-th iteration of the While loop from line 7 to line 14, the ordering π includes i elements, denoted by $x_1, x_2, \ldots, x_i$ where $x_1 = a$. We prove by induction on i that function VertexSorting eventually returns an ordering π that is either π' or the reverse of π'. For $i = 1$, $x_1 = a$ is an extremal vertex in π'. Assume that x_j and x_{j+1} appear consecutively in the ordering π', i.e., $|(\pi')^{-1}(x_j) - (\pi')^{-1}(x_{j+1})| = 1$, for all $j \leq i$. In line 8, set X includes two extremal vertices among all vertices that do not appear in π, according to π'. The two vertices in set X are thus uniquely defined, and include z and other vertex, call it b, which is in-between a and z in π'. Additionally, since x_j and x_{j+1} appear consecutively in the ordering π', for all $j \leq i$, it implies that b is adjacent to x_i in π'. At the end of this iteration, $x_{i+1} = b$ is appended to the tuple π.

It implies that any 1D-mobility temporal clique admits exactly two orderings π such that $\mathcal{R}(\mathcal{G})$ provides a 1D-mobility schedule from π.

3.4 Temporal Spanner

In this subsection, we show that any 1D-mobility temporal clique has a spanner of $\mathcal{G}$ of size $(2n-3)$.

Theorem 2. *Given a 1D-mobility temporal clique $\mathcal{G}$, let $\mathcal{H}$ be the temporal sub-graph of $\mathcal{G}$ consisting in the $(2n-3)$ edges that are adjacent with either v_1 or vertex v_n. $\mathcal{H}$ is a temporal spanner of $\mathcal{G}$.*

Proof. Let us consider the edge (v_1, v_n, t) that corresponds to the crossing of the initial two extremities on the line. When this happens, we have two sets: V_L (resp. V_R) corresponding to the agents being at the left of v_1 (resp. right of v_n) at time t. Before t, we got all edges of the form $v_1 v_l$ with $v_l \in V_L$ and of the form $v_r v_n$ with $v_r \in V_R$. After t, we get all edges of the form $v_1 v_r$ with $v_r \in V_R$ and of the form $v_l v_n$ with $v_l \in V_L$ (see Figure 2).

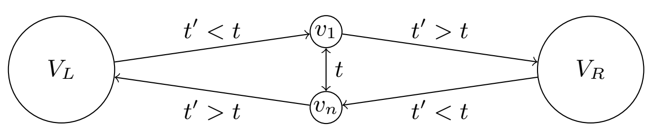

Fig. 2. Relative order of edge-labelling between the sets V_L, V_R and the two vertices v_1 and v_n. Edges are here to show how connectivity paths are used

As we have all edges connected to v_1 and v_n, we only need to prove that we keep connectivity between V_L and V_R, but also in between those sets:

- To connect a node $v_l \in V_L$ to $v_r \in V_R$, we use the path (v_l, v_1, v_r).
- To connect a node $v_r \in V_R$ to $v_l \in V_L$, we use the path (v_r, v_n, v_l).
- To connect a node $v_l \in V_L$ to $v'_l \in V_L$, we use the path (v_l, v_1, v_n, v'_l).
- To connect a node $v_r \in V_R$ to $v'_r \in V_R$, we use the path (v_r, v_n, v_l, v'_r).

4 Frozen Agents

We will consider a new case, where the temporal graph is no longer a clique, and introduce patterns with forbidden edges.

In this section, we consider the case where some agents are in a fixed position in the middle at the beginning. We partition the set of agents in three groups $A_1 \cup A_2 \cup A_3$. $A_1 = \{1, \ldots, k\}$,

$A_2 = \{k+1, \ldots, l\}$ and $A_3 = \{l+1, \ldots, n\}$. All agents of A_1 and A_3 will cross every other nodes. However, no pair of agents in A_2 will cross each other. At the end, agents end up in the order $n, n-1 \ldots, l+1, k, k+1, \ldots, l, k, \ldots, 2, 1$. We call the associated temporal graph a *frozen-agents-mobility temporal graphs*. Our graph is no longer a clique, as some pairs never cross.

In addition to the Patterns from Figure 1, we have to avoid the Patterns from Figure 3, where dotted edges represent edges that do not appear in the graph. The important element here is that if we have a dotted edge between two nodes, it means that both of them are frozen.

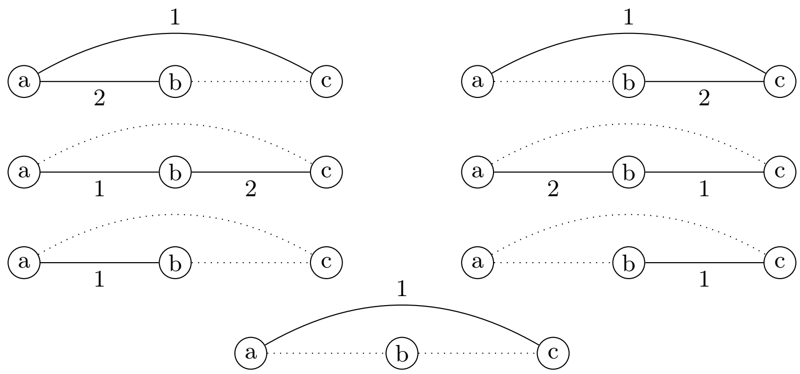

Fig. 3. Ordered forbidden patterns with forbidden edges in the 1D-mobility model with frozen agents. The ordering associated to each pattern is a, b, c.

Theorem 3. *An incremental single temporal graph is a frozen-agent temporal graph if and only if there exists an ordering of its nodes that excludes the ordered temporal patterns of Figures 3 and 1.*

As $\mathcal{G}$ is incremental and single valued, we assume without loss of generality that its lifetime is $T = n(n-1)/2 - (k-l)(k-l+1)/2$ and that λ is a bijection from E to $[T]$. Let $\mathcal{C}_{k,l,n}$ denote the class of incremental single temporal graphs with lifetime equal to the number of edges, i.e., λ is a bijection from E to $[T]$, which have an ordering excluding the ordered temporal patterns of Figure 3 and Figure 1.

Let us define $\mathcal{C}'_{k,l,n} \subseteq \mathcal{C}_{k,l,n} \times \mathcal{S}_n$ as the set of ordered incremental single temporal graphs $(\mathcal{G}, \pi)$ such that $\mathcal{G} = ((V, E), \lambda) \in \mathcal{C}_{k,l,n}$ and π is an ordering of V such that $\mathcal{R}(\mathcal{G})$ provides a frozen-agents-mobility schedule from π.

We first prove that any frozen-agents-mobility temporal graph is in $\mathcal{C}_{k,l,n}$:

Proposition 3. *For any frozen-agents-mobility schedule x of n agents in three groups $A_1 \cup A_2 \cup A_3$ producing a incremental single temporal graph $\mathcal{G}_x = ((V, E), \lambda)$, the initial ordering π of the agents excludes the patterns of Figure 3, and Figure 1.*

Proof. Lemma 1 gives us the proof for the patterns of Figure 1. About the patterns of Figure 3, we have the following observations. If we have two frozen nodes a and b such that a, b is a sub-sequence of π, a node from A_1 must meet a before b (top left pattern), and b before a if it is a node from A_3 (top right pattern). No node can start between two frozen nodes (patterns on the second line). Between three frozen nodes, we can only have dotted edges (the three patterns below).

We now show that forbidding these patterns fully characterizes frozen-agents-mobility temporal graphs. For that purpose, we construct a bijection h from $\mathcal{C}'_{k,l,n}$ to the set $R(w_{kln})$ of all reduced decompositions of $w_{kln} = n, n-1 \ldots, l+1, k, k+1, \ldots, l, k, \ldots, 2, 1$ is the permutation in $\mathcal{S}_n$.

Lemma 5. *Let $0 \leq k < l < n$ be three integers. Any temporal graph $\mathcal{G} \in \mathcal{C}_{k,l,n}$ having an ordering π excluding the patterns of Figure 1 and Figure 3, can be associated to a reduced decomposition $h(\mathcal{G}, \pi)$ of w_{kln}. Moreover, this reduced decomposition corresponds to a frozen-agents-mobility schedule and $\mathcal{G}$ is a frozen-agents-mobility temporal graph.*

Proof. Starting from the initial ordering $\pi_0 = \pi$, we define a sequence $\pi_1, \ldots, \pi_T$, where $T = n(n-1)/2 - (k-l+1)(k-l)/2$, of orderings, corresponding to what, we believe, to be the positions of the agents at each time step if we read the edges of $\mathcal{G}$ by increasing time label as a mobility schedule. More precisely, for each $t \in T$, π_t is defined from π_{t-1} as follows. Consider the edge $uv = \lambda^{-1}(t)$ appearing at time t in $\mathcal{G}$. We define τ_t as the transposition exchanging u and v in w_{t-1}. Equivalently, we set $\tau_t = (i, j)$ where i and j respectively denote the indexes of u and v in π_{t-1}, i.e. $\pi_{t-1}(i) = u$ and $\pi_{t-1}(j) = v$. We then set $\pi_t = \pi_{t-1}\tau_t$.

Our goal is to prove that $h(\mathcal{G}, \pi) := \tau_1 \cdots \tau_T$ is the desired reduced decomposition. For that, we need to prove that u and v are adjacent in $\pi_{t-1} = \pi_0\tau_1 \cdots \tau_{t-1}$ when uv appears at time t. For the sake of contradiction, consider the first time t when this fails to be. That is $\tau_1, \ldots, \tau_{t-1}$ are indeed adjacent transpositions, uv is the edge appearing at time t, and u, v are not consecutive in π_{t-1}. We assume without loss of generality that u is before v in π_0, i.e. u, v is a sub-sequence of π_0. We will mainly rely on the same observation we had for Lemma 2: For any pair of nodes x, y such that x is before y in π_0, x is before y in π_{t-1} if and only edge xy appears at t or later, i.e. $\lambda(xy) \geq t$.

The above observation implies that u is still before v in π_{t-1}. As u and v are not consecutive in π_{t-1}, there exists an element w between elements u and v in π_{t-1}. We have the following cases:

Case 1. w was already in-between u and v in π_0. As the relative order has not changed yet between these three nodes, we have the following possibilities:

- $\lambda(uw) > t$ and $\lambda(wv) > t$, contradiction with the patterns on the left of Figure 1;
- $uw \notin E$ and $\lambda(wv) > t$, contradiction with the second pattern of Figure 3;
- $vw \notin E$ and $\lambda(uw) > t$, contradiction with the first pattern of Figure 3;
- $uw \notin E$ and $vw \notin E$, contradiction with the last pattern of Figure 3.

Case 2. w was not in-between u and v in π_0. It means that exactly one edge between uw and vw have a time label smaller than t (if it was none or both, w would not be between u and v). We have the following possibilities:

- $\lambda(uw) < t$ and $\lambda(wv) > t$, contradiction with the top patterns of Figure 1;
- $\lambda(uw) > t$ and $\lambda(wv) < t$, contradiction with the bottom patterns of Figure 1;
- $\lambda(uw) < t$, $wv \notin E$ and w started before u, contradiction with the third pattern of of Figure 3;
- $\lambda(uw) < t$, $wv \notin E$ and w started after v, contradiction with the first pattern of Figure 3;
- $\lambda(wv) < t$, $uw \notin E$ and w started before u, contradiction with the second pattern of Figure 3.
- $\lambda(wv) < t$, $uw \notin E$ and w started after v, contradiction with the fourth pattern of Figure 3;

We get a contradiction in all cases and conclude that $\tau_1, \ldots, \tau_T$ are all adjacent transpositions. They thus correspond to a frozen-agents-mobility schedule x such that $\mathcal{G}_x = \mathcal{G}$.

Additionally, as each pair of elements occurs exactly in one transposition, $h(\mathcal{G}, \pi) = \tau_1 \cdots \tau_T$ has length $T = n(n-1)/2 - (k-l+1)(k-1)/2$ and we receive w_{kln}, i.e., $w_{kln} = \tau_1 \cdots \tau_T$, and the number of transpositions in $h(\mathcal{G}, \pi)$ equal to $l(w_{kln}) = n(n-1)/2 - (k-l+1)(k-l)/2$, so $h(\mathcal{G}, \pi)$ is a reduced decomposition of w_{kln}.

This concludes the proof of Theorem 3. There is no explicit formula to count the number of graphs in each class $\mathcal{C}_{k,l,n}$. However, there are ways to count them. More precisely, w_{kln}, is a *vexillary permutation*, since w_{kln} is $(2143) - \mathit{avoiding}$, see [17]. From [3], we can calculate the number of reduced decompositions of w_{kln} based on Hook length formula.

Thanks to [7], we are able to have more information on the class of graphs in which frozen-agent temporal graphs belong when we remove the labels on edges:

Proposition 4. *The set of frozen-agent temporal graphs contains threshold graphs and is contained in complement of proper interval graphs.*

Proof. First we can note that in the 7 patterns of Figure 3, the last five do not depend on the time labels, either by symmetry (3, 4) or because there is only one label on the pattern (5, 6, 7). Using [7], we know that the corresponding class of graph is the complement of proper-interval graphs.

Similarly, if we add the 2 first patterns ignoring their labels, this corresponds exactly to the particular case of threshold graphs [7].

5 Multi-Crossing Mobility Model

In this section, we consider schedules where a pair of agents can cross each other more than once.

In this scenario, forbidding pattern no longer works, as we can have multiple time values through λ. For this reason, to forbid a pattern, we will use the notion of *sliding windows* where we choose two time limits $T_1 < T2$, and the edges/crossing happening in this interval. Our patterns will still be single valued temporal graphs. To forbid a pattern of size k, we need that for any sliding window and any subgraph on k vertices where each edge has at most one appearance with λ restricted to this interval (noted $\lambda_{[T_1,T_2]}$), there is no isomorphism from the first graph to the second. We are removing the ordering restriction, as it was representing the starting order, knowing that each crossing would happen at most once.

After an arbitrary number of crossing, we can have any arbitrary ordering of the agents, and keep having crossings. Because of that, we cannot forbid any pattern from Figure 1. However, thanks to the observation of Lemma 1, we can exclude some pattern:

Theorem 4. *Any temporal graph associated to a 1D-mobility schedule must exclude the pattern of Figure 4 in any of its single-valued sliding window.*

Proof. Let assume that there exists some 1D-mobility schedule which produces a temporal graph where the pattern of Figure 4 on some nodes a, b, c and d. Thanks to Lemma 1 applied to 3 nodes, we know from the largest label which node is in between two others. We deduce that, at time T_1, b is between a and c, d is between b and c, and d in between a and b. There is no way

Recall that the order of edge-labelling of a triangle implies which agent is in the middle among these three agents at starting time T_1 of the time sliding window. For example, in figure 4, at time T_1, agent b is in the middle of a and c, agent d is in the middle of b and c, agent d is in the middle of a and b, which gives contradiction. Thus, the pattern is forbidden.

Even though we have a forbidden pattern, we cannot characterize this class only with single valued patterns. For example, we could not detect a subgraph with the pattern of Figure 4 where an edge is multiplied. More precisely, if we have $\lambda_{[T_1,T_2]}(cd) = \{5,6,7\}$ and $\lambda_{[T_1,T_2]}(bc) = \{8\}$, this cannot happen in a 1D-mobility schedule for the same reason, but it does not have any forbidden pattern in some subinterval. It brings out the question "what can be a good generalization of forbidden patterns on temporal graphs where edges have multiple times?"

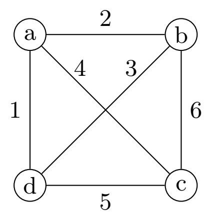

Fig. 4. A forbidden structure in multi-crossing mobility graphs.

6 Conclusion and Perspectives

In this paper, we have introduced the first notion of forbidden patterns in temporal graphs. In particular, this notion allowed us to describe a new class of temporal cliques corresponding to a mobility problem of agents crossing each other exactly once on a line. This new class of temporal cliques has spanners of size $2n-3$, following the conjecture from [5]. The mobility description allows the agents to adapt their speed to ensure that each crossing occurs in the correct order. A first open question is: can any 1D mobility temporal clique be the result of crossings if the agents move at constant speed, choosing wisely the distance at which they start? We can note that, for each crossing to occur from a starting situation, we would need to sort the agents by increasing speed from left to right.

Another question that arises is: can we find a mobility model on more dimensions that also provides a temporal clique that could be studied?

Our patterns only consider single times on the edges. One perspective is to figure out how to describe forbidden patterns on edges such that λ provides more than one time slot. Considering sub-intervals where λ gives at most one value is a possibility, but we have seen that in the case of multiple crossings this does not seem to allow to describe all forbidden patterns. This raises another question: is there a way to fully describe multi-crossing mobility model with forbidden patterns?

Our work can also be seen as a characterization of square integers matrices in terms of patterns, perhaps it could be generalized to a study of well-structured matrices as in the seminal work of [13] on Robinsonian matrices which are closely related to interval graphs.

Acknowledgement This work was supported by ANR project TEMPOGRAL (ANR-22-CE48-0001). The second author has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 945332.

References

1. Awerbuch, B., Even, S.: Efficient and reliable broadcast is achievable in an eventually connected network. In: Proceedings of the third annual ACM symposium on Principles of distributed computing. pp. 278–281 (1984)
2. Bandlow, J.: An elementary proof of the hook formula. the electronic journal of combinatorics pp. R45–R45 (2008)

3. Billey, S., Pawlowski, B.: Permutation patterns, stanley symmetric functions, and the edelman-greene correspondence. In: 25th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2013). pp. 205–216. Discrete Mathematics and Theoretical Computer Science (2013)
4. Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N.: Time-varying graphs and dynamic networks. IJPEDS **27**(5), 387–408 (2012)
5. Casteigts, A., Peters, J.G., Schoeters, J.: Temporal cliques admit sparse spanners. Journal of Computer and System Sciences **121**, 1–17 (2021)
6. Damaschke, P.: Forbidden ordered subgraphs. Topics in Combinatorics and Graph Theory: Essays in Honour of Gerhard Ringel pp. 219–229 (1990)
7. Feuilloley, L., Habib, M.: Graph classes and forbidden patterns on three vertices. SIAM Journal on Discrete Mathematics **35**(1), 55–90 (2021)
8. Frame, J.S., Robinson, G.d.B., Thrall, R.M.: The hook graphs of the symmetric group. Canadian Journal of Mathematics **6**, 316–324 (1954)
9. Guzmán-Pro, S., Hell, P., Hernández-Cruz, C.: Describing hereditary properties by forbidden circular orderings. Appl. Math. Comput. **438**, 127555 (2023)
10. Hell, P., Mohar, B., Rafiey, A.: Ordering without forbidden patterns. In: Algorithms-ESA 2014: 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings 21. pp. 554–565. Springer (2014)
11. Holme, P., Saramäki, J.: Temporal networks. Physics reports **519**(3), 97–125 (2012)
12. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. In: Proceedings of the thirty-second annual ACM symposium on Theory of computing. pp. 504–513 (2000)
13. Laurent, M., Seminaroti, M., Tanigawa, S.: A structural characterization for certifying robinsonian matrices. Electron. J. Comb. **24**(2), 2 (2017)
14. Michail, O.: An introduction to temporal graphs: An algorithmic perspective. Internet Mathematics **12**(4), 239–280 (2016)
15. Skrien, D.J.: A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs. Journal of graph Theory **6**(3), 309–316 (1982)
16. Stanley, R.P.: On the number of reduced decompositions of elements of coxeter groups. European Journal of Combinatorics **5**(4), 359–372 (1984)
17. Tenner, B.E.: Reduced decompositions and permutation patterns. Journal of Algebraic Combinatorics **24**, 263–284 (2006)
18. Villani, N.: Dynamic Unit Disk Graph Recognition. Master's thesis, Université de Bordeaux (2021), `https://perso.crans.org/vanille/share/satge/report.pdf`
19. Villani, N., Casteigts, A.: Some thoughts on dynamic unit disk graphs. Algorithmic Aspects of Temporal Graphs IV (2021), `https://www.youtube.com/watch?v=yZRNLjbfxxs`, satellite workshop of ICALP