

# TEMPOGRAL: TEMPORal GRaph ALgorithms: Models, Questions, Resolution

ANR Proposal 2022

Fondements du numérique

The table below provides the list of people involved in TEMPOGRAL. For each of them, we give the current position, the person-months workload, the themes in which they will be involved, and finally their role in the project.

Partner	Name	First Name	Position	Involv.	Themes	Coord.
LaBRI/ U. Bordeaux	<b>Casteigts</b>	<b>Arnaud</b>	<b>MCF HDR</b>	<b>30</b>	C1,C2,T1,T2,A2,A3	<b>Global</b>
	Corsini	Timothée	PhD std (MESR)	25	C2,T3,A3	
	Gavoille	Cyril	PR	16	C2,T3,A1	
	Hanusse	Nicolas	DR CNRS	15	C1,C3,T1,A1	
	Klasing	Ralf	DR CNRS	18	C1-C3,T1,A3	
	Johnen	Colette	PR	16	C3,T2,A3	
			PhD student	36	C2,T1,T2,A3	
LITIS/ U. Le Havre Normandie	Balev	Stefan	MC	16	C3,C4,A2,A3	
	Guinand	Frédéric	PR	11	C1,C4,A2,A3	
	Pigné	Yoann	MC	11	C2,C4,A2,A3	
	<b>Sanlaville</b>	<b>Eric</b>	<b>PR</b>	<b>24</b>	C1,C3,T1,T2,A3	<b>Local</b>
	Vernet	Mathilde	MC	12	C1,C3,T2,A3	
			PhD student	36	C1,C3,A3	
IRIF/ U. Paris Cité	Habib	Michel	PR Em.	15	C2,T2,T3,A3	
	de Montgolfier	Fabien	MCF	18	C2,T1,A3	
	Naserasr	Reza	CR CNRS	15	C1,C3,T1,T3,A3	
	Rabie	Mikaël	MCF	20	C1,T2,A1,A3	
	<b>Viennot</b>	<b>Laurent</b>	<b>DR INRIA</b>	<b>27</b>	C2,C4,T2,A1,A3	<b>Local</b>
			Post Doc	12	C2,T1,T2,A1	
		Post Doc	12	C1,T1,T2,A3		

The citations in this document appear in red bold font when they involve at least one member of the project (e.g. [XX]).

## 1 Proposal’s context, positioning and objectives

### 1.1 Context

Graphs are a fundamental modeling tool in science. They have been used for modeling phenomena in fields ranging from statistical physics to communication networks, distributed algorithms, logistics, biology, medicine, and social networks. Despite great successes in these areas, many real-world phenomena are dynamic and fall beyond the expressivity of standard graphs. In the past two decades, increasing interest has been devoted to temporal graphs (also called time-varying, time-dependent, evolving, or simply dynamic), in which the presence of edges and (sometimes) nodes depends on time. A basic example of temporal graphs is shown in Figure 1. In this example, time is discrete, more sophisticated settings exist (see e.g. [36], [16]).

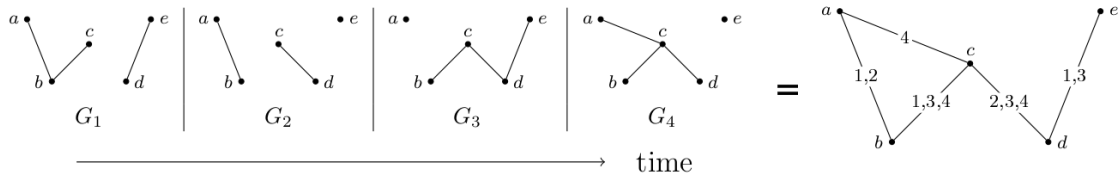


Figure 1: Two possible ways of representing the same evolution: As a sequence of “snapshots” (left), and as a graph whose labels encode presence times (right).

Early studies of temporal graphs occurred, independently, in each of the above fields. This led to a proliferation of new concepts and algorithmic questions, some of which are specific to a domain, while others are general, yet discovered independently and under various names. A typical example is the one of *temporal paths*, which are paths that can be realized over time (along ascending labels), also known as *time-respecting paths* or *journeys*.

The aim of our project is to develop a fundamental, i.e. *domain-independent*, theory of temporal graphs, with a focus on characterizing algorithmic problems and tools which are intrinsically temporal. The project has three main axes, which are (1) Classification of problems, (2) Tractability aspects, and (3) Algorithmic techniques.

## 1.2 Main themes

The project focuses on three main *axes* (semantically equivalent to workpackages), namely “Classification of temporal problems” (C), “Tractability of temporal problems” (T), and “Algorithmic tools” (A). Each axis consists of three or four interconnected research *themes* (semantically equivalent to tasks):

<b>C1.</b> What makes a problem intrinsically temporal? How to characterize temporality?	p 3
<b>C2.</b> Problems defined by the search for temporal substructures	p 3
<b>C3.</b> Temporal versions of optimization problems on graphs	p 4
<b>C4.</b> Problems related to the generation of temporal graphs	p 5
<b>T1.</b> Restricting the snapshot or the underlying graph and corresponding parameters	p 6
<b>T2.</b> Hierarchies of temporal graph classes	p 7
<b>T3.</b> Forbidden temporal patterns vs. tractability	p 8
<b>A1.</b> Design of advanced data structures	p 10
<b>A2.</b> Generation of temporal graphs	p 11
<b>A3.</b> Specifically designed algorithms for temporal problems	p 12

To summarize the objectives of this project, it is about proposing a global and integrated vision of the new questions arising on temporal graphs. The construction of a typology of problems, the proposal of new analysis tools for the design of algorithms, will pave the way for the building of a theory of temporal graphs. Not all of the knowledge can be produced within the framework of this project, but we hope that it will significantly contribute to its development.

## 1.3 Classification of temporal problems (C)

Algorithmic problems can be classified in various ways. The most common is certainly in terms of their computational complexity, i.e. the amount of resources an algorithm uses to solve the problem. Another, perhaps more qualitative way, is in terms of the expressivity required to formulate the problem, e.g. first-order or second-order logic. In the case of temporal problems, another natural question is the extent to which the problem is intrinsically temporal. The question is not so simple, because many of the temporal problems that matter in application scenarios are ultimately reducible to standard (i.e. non-temporal) NP-hard problem. Thus, what makes a problem truly temporal is not clear. We will explore different ways of answering this question.

**Theme C1:** *What makes a problem intrinsically temporal? How to characterize temporality?*

Some problems are more temporal than others. Consider a typical graph problem like finding a maximum (or maximal) dominating set. Given a temporal graph  $\mathcal{G}$ , one may reformulate this problem as finding a set of vertices  $S$  such that all vertices in the graph are dominated by a vertex of  $S$  in (i) at least one snapshot, or (ii) in all snapshots. (See Section 1.4 of [15] for a general discussion on such adaptations.) It turns out that the former problem amounts to computing a standard dominating set in the *union* of all snapshots, while the latter is strongly related to computing one in the *intersection* of all snapshots. In other words, these problems are naturally reducible to instances of *static* graph problems. For other problems, e.g. for temporal flows [49], reductions also exist to static graph problems (e.g. using the concept of time-expanded graph); however, the transformations are more costly in time and space. For yet other problems, such as problems related to temporal reachability (discussed later on), no natural reduction to static graph problems have yet been identified. What does make these problems different? Can these differences be characterized, leading to several types (or degrees) of temporal graph problems?

Beyond the scale of the transformation from temporal problems to static problems, qualitative properties of a problem may indicate the degree to which it is temporal. In the above example, where the solutions are defined in the union or the intersection of the snapshots, one can observe that these solutions remain stable under reshuffling of the snapshots – i.e. the order of the events does not matter. Other ways of adapting the same kind of static problems have been considered, resulting in less straightforwardly reducible problems. For instance, if the solution is allowed to evolve over time, though in a restricted way, this creates interesting dependencies between the snapshots in which the order of events does matter. In reconfiguration problems, for instance, the goal is to produce a sequence of elementary changes to go from one solution to another solution (see e.g. [12]). The aim is to figure out what are the global optimization goals and modification constraints that make such problems temporally interesting, and connect this literature to existing temporal reformulations of certain problems (possibly through new reductions).

**Objectives:**

1. Make an inventory of existing reductions from temporal problems to static problems.
2. Identify abstract properties of problems that hint at its temporality.  
(E.g. the property that solutions are not invariant under reshuffling of the snapshots.)
3. Define various classes of reductions from temporal to static problems.  
(E.g. is the transformed instance the same size as the original? Can the problem be reduced to a static problem on the same set of vertices? To reconfiguration problems?)

**Related themes:** C2,C3,T1,T2,A1,A3

**Theme C2:** *Problems defined by the search for temporal substructures*

An entire class of static graph problems consists of searching for specific substructures in a graph. An archetypal example in this class is that of computing a spanning tree (minimum or not, distributed or centralized). In temporal graphs, several analogs of such problems can be defined, a possible approach being to formulate the question with respect to the snapshots or to the underlying graphs. However, more interesting phenomena seem to occur when the substructure itself involves temporal concepts. In their seminal paper, Kempe, Kleinberg, and Kumar [39] asked the following question:

“Given a temporally connected network  $G = (V, E)$  on  $n$  nodes, is there a set  $E' \subseteq E$  consisting of  $O(n)$  edges so that the temporal network on the subgraph  $(V, E')$  is also temporally connected? In other words, do all temporal networks have sparse subgraphs preserving this basic connectivity property?”

In this question, they consider a model where each edge has a single time label, but the discussion is more general. Essentially, they ask whether temporal analogs of spanning trees, or more generally, temporal analogs of sparse spanners always exist in temporal graphs. Surprisingly, after nearly two decades, the answer turned out to be negative: some temporal graphs use  $\Theta(n^2)$  labels for their temporal connectivity, and all of these labels are necessary [6]. Guaranteeing the existence of sparse spanners (i.e. spanners of size  $o(n^2)$ ) requires severe restrictions, such as a complete underlying graph [19]. An example of a minimum temporal spanner is shown in Figure 2.

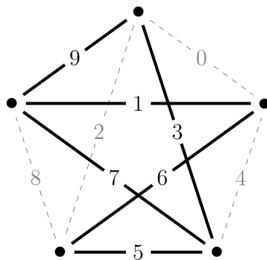


Figure 2: Minimum temporal spanner in a complete temporal graph (with single time labels).

The above example shows that searching for temporal substructures, even natural ones, may lead to unexpected results. Another example is the search for maximum temporal components [11], which turns out to be a problem very related to finding maximum cliques in static graphs. We will investigate spanner problems and other problems involving the search for particular substructures, trying to extract from these investigations a number of common knowledge (facts, reductions).

Doing so, we will consider other typical structures searched in static graphs and make a list of the ones that have already been considered in a temporal setting, such as Hamiltonian and Eulerian cycles [44]. Note that for cliques and independent sets, the natural extensions lead to solving the static version on the union of all snapshots whereas the direct extension to spanning trees leads to work on the intersection graph. Some of the questions will be highly related to the problem of temporal subgraph isomorphism, which has been considered for some specific applications so far, like modeling car traffic at an intersection, or strategies in a collective sport (see e.g. [45]). In the latter case, one may be interested in identifying a given sequence of passes during some given time interval, i.e. a temporal pattern. The definition of temporal patterns is also discussed in item T3.

**Objectives:**

1. Identify interesting temporal substructures.
2. Investigate spanner problems.

**Related themes:** C1,C3,C4,T1,T3,A1,A2,A3

**Theme C3:** *Temporal versions of optimization problems on graphs*

This theme focuses on optimization problems. Shortest paths and flow problems, among others, have natural counterparts in temporal graphs, and have already been studied. However, some open questions remain; for example, is it possible to solve some flow or path problems without using time expanded graphs? Generally speaking, problems based on paths (as TSP, VRP) have often a quite direct temporal version and some results can be found in the literature, see for instance [44] for the TSP. Interestingly, various shortest paths problems can be naturally defined in temporal graphs for various definitions of lengths: length can be defined in terms of duration of the temporal path, its number of edges, or its sum of costs when each temporal edge is associated to a cost. Is it always possible to reduce such computations to shortest paths in the time expanded graph? As

a follow-up to our work on providing a generic algorithm for solving the shortest temporal paths problem for an algebraic definition of cost [13], we would like to investigate if this framework can be extended to temporal walks with restricted waiting at nodes, a setting where the computation of temporal paths becomes NP-hard [18] while temporal walks can still be computed in polynomial time [10].

For other optimization problems, defining the temporal version is not immediate. For instance, the static Steiner tree problem in its simplest version arises from the wish to connect a fixed subset of vertices, using a minimal number of additional vertices. In a temporal graph, one may wish to keep that property for each snapshot. Using always the same set of edges is impossible. On the other side, computing a new Steiner tree for each snapshot might be unwise, as it produces potentially very different solutions at each time. One first interesting variant consists in computing a Steiner set (set of vertices) of minimum size, allowing connectivity at each snapshot through varying edge sets. Another variant is to look for successive solutions that differ by a minimum number of changes (note the closeness here to reconfiguration problems already mentioned).

Hence, it might be fruitful to investigate systematic ways to build temporal counterparts and, according to these methods, to identify problems that might admit some specific algorithms, more efficient (in time and/or space) than the direct use of the classical ones.

**Objectives:**

1. Study problems that results from considering temporal paths instead of paths.
2. Make an inventory of temporal versions of most classical optimization problems.
3. Identify canonical ways of (re)defining problems, beyond temporal paths.  
(E.g. by considering a sliding window)

**Related themes:** C1,C2,T1,T2,A3

**Theme C4:** *Problems related to the generation of temporal graphs*

We will consider various generative models of temporal graphs. Some of these models are already identified, including:

**Evolutionary models:** What are the interesting evolution models? A first aspect of this question concerns models where elementary operation are performed on a snapshot to obtain the next one. This typically includes random Markovian models where random edge insertions and deletions are performed based on the current snapshot and two parameters (birth rate, death rate), see e.g. [24]. The graph may also result from the evolution of another underlying model. A typical example comes from moving particles, objects or agents in some metric space where the edges at a given time depend on proximity. One may generate temporal graphs that satisfy desired properties by choosing carefully how the agents are moving [46].

**Temporalisation:** Given a static graph, assign time labels to its edges. A basic example is a temporal clique which is obtained by assigning one time label to each edge of a clique as studied in [20]. Another typical example comes from network design problems where interactions are known in advance but where there is some degree of freedom or uncertainty on when they can occur. For example, in a public transport network, the operator can change the departure time of the vehicles to obtain a temporal graph with better temporal connectivity.

**Random edge ordering:** In [21], a random model of temporal graphs with only two parameters  $n$  and  $p$  is defined as follows: (1) pick a random Erdős-Rényi using the same parameters, (2) decide of a random permutation of the existing edges, (3) interpret the rank in this permutation as the presence time.

We will study what are the interesting graph properties that are guaranteed by these main generative models, in a way similar to static graphs, see e.g. [32] for a recent attempt to classify random generators according to the satisfaction of various static graph properties. We think of natural properties for solving main problems, such as temporal connectivity, stating that each node is temporally connected to each other node, which seems a natural assumption when considering many problems. For example, it was shown in [21] that for sufficiently large  $n$ , the graph becomes temporally connected at  $p = 3 \log n/n$ . We will also consider the generation of graphs satisfying deterministic properties related to the tractability of problems (related to theme T1), as well as the algorithmic problem of the generation itself (theme A2). Indeed, the latter question becomes central when considering the exhaustive enumeration of all temporal graphs in a certain class, or the generation of an instance chosen uniformly in this class.

**Objectives:**

1. Identify main generative models of temporal graphs.
2. Classify generative models with respect to the temporal graph properties they guarantee.
3. Find interesting temporal graph classes that can be generated exhaustively or randomly.

**Related themes:** C2,T2,T3,A2

## 1.4 Tractability of temporal problems (T)

The search for efficient algorithms to solve temporal problems often comes up against difficulties due precisely to the temporal dimension. For example, it was shown in [38] that deciding whether  $k$  vertex-disjoint temporal paths exist between two vertices is NP-complete in temporal graphs, whereas the analog problem is tractable in static graphs. In static graphs, the tractability of this problem is due to a duality between maximum flows and minimum cuts that breaks in temporal graphs. Similarly, computing a maximum connected component based on temporal paths is as hard as finding a maximum clique in static graphs [11], mainly due to the fact that temporal components do not form equivalence classes, in contrast to their static analogs. Since then, many other tractable (or approximable) static problems have been shown to have intractable (or inapproximable) analogs in temporal graphs (see e.g. [2, 17, 29, 33]). (A notable counter-example is the case of flows, see for instance [49].) The main goal of this part is to understand the typical source of intractability in temporal problems, and to identify various ways to circumvent it.

**Theme T1:** *Restricting the snapshot or the underlying graph and corresponding parameters*

A natural way to make a (difficult) graph problem tractable is by restricting the class of graphs that the algorithm addresses (trees, planar graphs, etc.). Such restrictions also impact the very existence of solutions; for example, no  $o(n^2)$ -sparse spanners are guaranteed to exist in general temporal graphs [6], but they do if the underlying graph is complete [19] (incidentally, they also do if the time labels are random [21]). Similarly, exploration by mobile agents is facilitated if the snapshots are all connected and the underlying graph is a cactus [37]. A more subtle way of analyzing tractability is to characterize the way certain graph parameters impact the computational complexity of a considered problem, with a focus on those parameters whose fixed value causes the problem to be tractable. Typical parameters include treewidth, feedback vertex set, and cliquewidth, among many others (see [47] for a hierarchy). It is natural to apply this approach in temporal graphs by restricting the snapshot or the underlying graph. This approach has been considered, for example in [31], to characterize the tractability boundary of separation problems in temporal graphs. In [18], it was considered for the following basic problem: given two vertices  $u$  and  $v$  and a certain duration  $\Delta$ , is there a temporal path from  $u$  to  $v$  that pauses at most  $\Delta$  units of time at each intermediate node (*a.k.a.* a  $\Delta$ -restless path). This problem is NP-hard

even under severe restrictions (e.g. when the lifetime is only 3). However, it is fixed-parameter tractable (FPT) for natural parameters like treedepth or feedback edge number. The tractability boundary is illustrated in Figure 3.

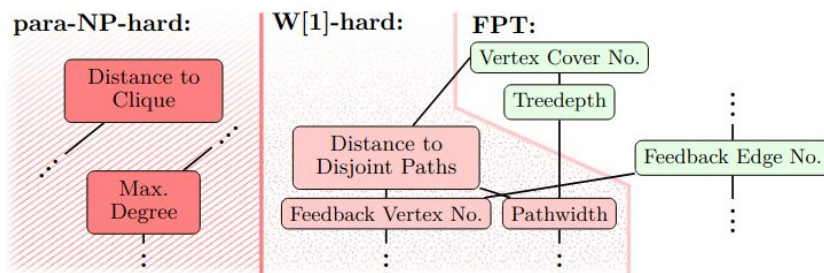


Figure 3: Tractability of temporal paths with bounded waiting times, in terms of parameters of the underlying graph. (Picture from [18].)

We will study such boundaries for a number of basic temporal graph problems. In doing so, we will try to understand what parameters are more inherently related to the (in)tractability of temporal problems. Another goal is to define parameters which are themselves temporal. In [18], a first step towards this goal was made by introducing a new parameter called “timed feedback vertex number”, which counts the number of vertex *appearances* that have to be removed for the underlying graph to become cycle-free. While it is finer, this parameters still does not fit all standards discussed in C1 for being truly temporal (e.g., its value remains invariant under reshuffling of the snapshots). We will design parameters of a temporal nature and investigate their impact on basic temporal problems. For example, the number of temporal cycles may play a role in the above problems. Another temporal parameter of interest could be the number of temporal paths that one can use over disjoint time intervals between any two vertices (i.e. a measure of how many times one can go back and forth between these vertices).

**Objectives:**

1. Find how tractability can arise from restricting the snapshots or the underlying graph.
2. Identify new temporal parameters and their impact on difficult problems.

**Related themes:** C1,C2,C3,T2,T3,A1,A3

**Theme T2:** *Hierarchies of temporal graph classes*

Previous attempts at defining temporal restrictions that make a problem tractable were made in the context of distributed network algorithms. While the set of communication links in many types of networks (e.g. vehicular networks, sensor networks) often varies in unpredictable ways, they often satisfy weaker types of properties over time and space. For example, the network may never be connected at a given instant, and yet offer temporal reachability between all pairs of nodes (i.e. temporal connectivity). The network may also offer that all pairs of nodes interact at least once, or recurrently, or that the interactions are periodic, etc. All these properties impact the feasibility and tractability of temporal problems. Furthermore, the corresponding classes of temporal graphs can be related through inclusion, leading to a hierarchy illustrated in Figure 4. (More details in [16, 15].)

We will study the impact of such properties on (non-distributed) temporal graph problems and characterize their impact on tractability. Observe that many of the above classes are defined over an infinite lifetime, which is very appropriate in a distributed context (long-lived networks), but less

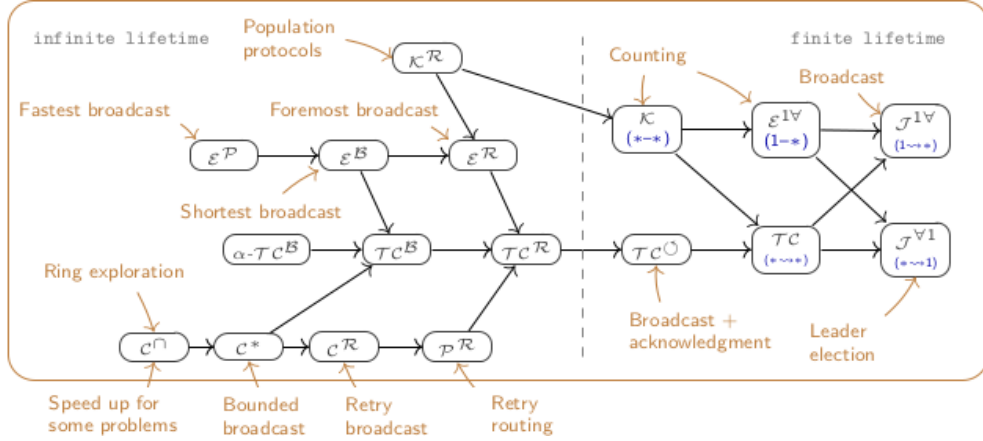


Figure 4: Hierarchy of temporal graph classes, in relation to various distributed tasks.

common in centralized algorithmics, where the input is usually finitely represented. Two directions are envisioned here: (1) looking at temporal graph problems whose input is discovered in an online fashion (e.g. [22]), and (2) looking at infinite temporal graphs which are finitely represented, e.g. periodic temporal graphs or graphs generated by finite computational objects (e.g. automata and algorithms). It is expected that the above properties will impact the tractability of algorithmic problems in a similar way as for distributed algorithms. Doing so, we will also define new classes of temporal graphs which impact the tractability of problems in various ways. We will also continue to study the impact of these classes from the point of view of distributed computing (and self-stabilization, as done recently in [4]). For example, the last parameter mentioned in T1 (number of back and forth temporal paths) could define an infinite hierarchy of classes of temporal graphs, whose levels are related to specific distributed tasks, e.g. broadcasting in level 1, broadcasting with acknowledgement in level 2, etc.

**Objectives:**

1. Differentiate online versus offline tractability.
2. Study periodic temporal graphs and finitely represented infinite temporal graphs.
3. Identify hierarchies of temporal graph classes in the spirit of distributed classifications.

**Related themes:** C1,C3,C4,T1,A2

**Theme T3:** *Forbidden temporal patterns vs. tractability*

Many graph classes can be defined in terms of forbidden patterns. A famous example is the class of planar graphs which are the graphs excluding  $K_5$  and  $K_{3,3}$  as a minor. Such structure can typically be exploited to obtain better tractability. A first track of research consists in finding natural temporal versions of patterns used in static graphs. Topological minors could be a promising target: as they can be defined in terms of paths, it is natural to consider a similar structure with temporal paths. However, this naive approach hits the asymmetry of time: time intrinsically induces an asymmetry (even with symmetric edges) which seems incompatible with the classical minor theory which concerns inherently undirected graphs. However, current work on extending minor theory to directed graphs could shed light on this approach. Another challenging task is then to identify problems where tractability improves when restricting the problem to temporal graphs excluding such a temporal minor.



As an orthogonal direction, we will consider patterns related to some ordering of the vertices. Indeed, many hereditary classes of graphs can be defined this way: a graph is in the class when it admits a total ordering of the vertices avoiding a set of edge or non-edge patterns with respect to the ordering, see [30] for an extensive study of all elementary patterns. Equivalently, such an ordering can be seen as a certificate that the graph belongs to the class. Moreover, computing it is often inherent in efficient recognition. In the temporal context, it seems natural to suppose that some sequences of edges are forbidden. The time ordering is then the important dimension. Do the characterizations developed for the vertex dimension can help understand patterns defined on the temporal dimension? Alternatively, can we define temporal graph classes by playing on both dimensions considering graphs where there exists a vertex ordering avoiding some temporal patterns of edges? Figure 5 illustrates some possible patterns.

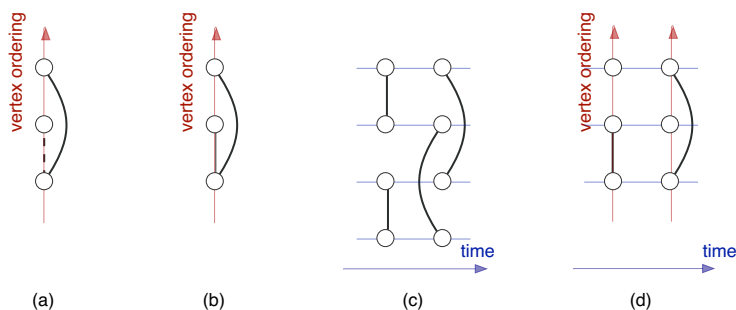


Figure 5: Forbidden patterns: (a) Interval graphs have a vertex ordering excluding this static pattern (the dashed line represents the absence of the edge). (b) Forests have a vertex ordering excluding this pattern. (c) Excluding this temporal pattern restricts the structure of temporal spanners (two nodes on the same line represent the same node at two distinct instants). (d) A temporal pattern using both a vertex ordering and time; excluding it implies some kind of topological ordering with respect to temporal paths.

Another approach in this theme is to study patterns that can naturally arise in practical temporal graphs or networks. For example, a public transit network can naturally be represented by a temporal graph where nodes correspond to bus stops and where a temporal edge represents an elementary movement of a bus from a stop to the next one in its trip. Such networks mostly satisfy a symmetry property: for each sequence of edges corresponding to a trip of a bus, there often exists a symmetric sequence of edges visiting the same nodes in reverse order as a bus line generally has buses circulating in both directions. Does such symmetry have tractability consequences?

**Objectives:**

1. Identify forbidden patterns enabling better tractability.
2. Investigate elementary edge patterns related to time ordering and possibly an additional vertex ordering.
3. Find interesting patterns inspired from practical temporal networks.

**Related themes:** C2,C4,T1,A1,A2,A3

## 1.5 Algorithmic tools (A)

The third part of this project will be devoted to positive results and algorithmic techniques, with a broad perspective. This objective will interact strongly with the other parts; in particular, in

identifying how restrictions on the input can be exploited algorithmically. We will also develop a set of tools and techniques, with the same underlying motivation as in the entire project, that is, focusing on techniques of ubiquitous uses in temporal graph problems.

Doing so, we will also develop algorithms for the generation of temporal graphs satisfying certain properties (e.g. for experimentation or conjecture testing). Preliminary work in this direction by some of the participants include [48] (spatio-temporal graphs) and [21] (random temporal graphs satisfying certain reachability properties). The generation of a temporal graph can also be seen as an inverse problem: given a static graph and some scheduling constraints on the edges, how to associate a presence time with each edge so as to guarantee certain properties, such as a good temporal connectivity for example? Finally, we will revisit, extract, and attempt to generalize techniques developed in the recent literature on temporal graphs, as well as in neighbor fields like re-optimization (a.k.a. “dynamic” graph algorithms) and reconfiguration problems. The project will also benefit from the experience of some of its members in tackling optimisation problems on temporal graphs (e.g., Pareto optimal paths [13], maximal persistent connected components [50], minimum cost flows [49]) and in distributed reconfiguration [23].

**Theme A1:** *Design of advanced data structures*

Several data structures are classically used to represent a temporal graph: sequences of graphs, sequences of links [42], sequences of events [27], and ordered adjacency lists [14], to name a few. One representation can generally be transformed into the other in polynomial time under mild assumptions.

It is interesting to notice that these representations are generally sorted according to time. While this seems a minor assumption as an unsorted representation could be sorted within a logarithmic factor from linear time, it becomes important when considering problems that can be solved in near linear time. A first study will consider the benefit of sorted data structures. For example, when do they allow to save a logarithmic factor?

A main study in this theme will focus on answering basic queries beyond what basic data structures enable. We are thinking in particular of distance queries, reachability queries, or adjacency queries. In the context of static graphs, there is a vast literature around oracles and labeling schemes about computing advanced data structures enabling fast answers to such queries. Some participants of the project have strong expertise in this domain, see e.g. [26, 40, 41]. For example, a distance oracle is a data structure allowing to answer distance queries between two nodes. A crucial point in its design is the trade-off between space and time: the size of the data structure versus its query time. An elegant and efficient way of building such a distance oracle consists in assigning a short bit-string label to each node such that the distance between two nodes can be obtained from their labels. This is what is called a distance labeling scheme. There is a vast literature studying the possible theoretical trade-offs of such data structures as well as more recent works on the efficiency of distance oracles in practice, see e.g. [7].

Adapting such techniques to temporal graphs is still a wide open area. A first seminal work in that direction is [1] where a distance labeling scheme is maintained dynamically as edges are added to an evolving graph. Its main intent is to maintain dynamically distance labels of the nodes of the current graph in the incremental model (where edges and nodes are added but never deleted), and many follow-up works followed that direction. Such work does not address distances with respect to temporal connectivity. Interestingly, [1] supports historical queries reporting the evolution of the distance between two nodes from snapshot to snapshot. A similar example is that of kinetic data structures [9, 34], where the adjacency information is updated as efficiently as possible after the nodes move in the euclidean plane (based on proximity). However, addressing intrinsically temporal queries is still beyond such approaches.

A first step in that direction is made in [25] where a labeling scheme for (temporal) reachability queries is proposed. It is based on using the classical 2-hop labeling approach (see e.g. [7, 40, 41]) on the time expanded graph representation of the temporal graph. The data structure indeed solves a more difficult problem: it allows to test whether a node  $v$  is reachable before time  $t'$  when starting from a node  $u$  at time  $t$ . This leaves open the possibility of more efficient approaches for

basic reachability over the whole temporal graph history, and more generally to design labeling schemes that encapsulate the temporal dimension more globally. Moreover, such experimental works do not provide any theoretical insight concerning their efficiency in terms of memory usage or query time. We believe that particular temporal graph structures could allow such guarantees as it is the case in static graphs, see e.g. [26, 40].

**Objectives:**

1. Identify the benefit of sorting in basic data-structures.
2. Find labeling schemes for basic temporal queries such as reachability.
3. Study spatio-temporal data structures (e.g. considering physical movements).

**Related themes:** C1,C2,C3,T1,T3,A3

**Theme A2:** *Generation of temporal graphs*

Generating temporal graphs is a multi-facet and difficult question. As stated for Theme C4, a natural objective is to generate a set of instances (possibly a singleton, possibly the complete set), satisfying some temporal property, either deterministically or randomly. It seems natural, and indeed it is usually the case, to consider sequential or incremental mechanisms (one snapshot graph being obtained from the previous one). Many random generators have been proposed, usually to build static graphs. First the Erdős-Rényi model, and later preferential attachment based models, see [3, 5], build a sequence of graphs with increasing number of edges and vertices. Usually, the process is stopped when reaching a given number of vertices. Preferential attachment guarantees a scale-free property (that is the degree distribution follows a power law).

When looking for temporal graph generation, the first and quite natural kind of mechanisms builds a temporal graph with a constant number of vertices. It starts from the underlying graph, which may or may not be complete. It then iteratively and independently adds or removes edges from this graph, according to some random mechanism like Markov chains, see [24, 50]. This method allows one to guarantee a kind of stability in the snapshot graph size. However, because of the independence of the evolution of edges, this mechanism does not by itself guarantee the persistence of some global properties, even if the underlying graph satisfies it. One may try to obtain this guarantee by other means. [51] proposes a memory of the links based mechanism for social networks. Other processes may use spatial features associated to vertices: those have a location and an edge exists if the spatial distance between two vertices is less than a threshold (random geometric graphs, see [28]). The temporal feature is obtained, for instance, by the birth or death of a vertex according to its neighborhood (rules similar to those used in cellular automata), or by moves performed by the vertices, inspired by communication networks between mobile sensors, see [46]. In the first case, the challenge is to maintain a relative stability of the network (avoid explosion or disappearance of the vertices) by identifying the generator parameter values leading for instance to some phase change effects, coupled with a given variability of the temporal graph. Patterns as clusters, specific subgraphs, may also appear which are difficult to anticipate by the random mechanism alone. In the second case, it leads to identify the simplest mobility models allowing to satisfy a given property.

Our goal in this theme is the search for temporal graph generators that will satisfy, maybe only with high probability, some of the temporal properties that other themes identified. Some examples are also mentioned in Theme C4. In this theme, we will mainly focus on mechanisms based on spatial features, either inspired by cellular automata, or by movement patterns of underlying agents. Another, perhaps more exploratory question is the search for non incremental generation mechanisms that may be more appropriate for satisfying desired temporal properties.

**Objectives:**

1. Study generation mechanisms that induce given temporal properties.
2. Consider these models from the standpoint of enumeration or uniform sampling.
3. Investigate non incremental generation mechanisms.

**Related themes:** C2,C4,T2,T3**Theme A3:** *Specifically designed algorithms for temporal problems*

The ambition of this theme is the design of new algorithms and techniques that are relevant to various types of temporal graph problems (in the sense of Theme C1). To achieve this goal, we will first survey existing techniques in classical algorithmic topics, which seem appropriate for *some* types of temporal problems. For example, the field of re-optimization algorithms (also called multi-stage algorithms, or sometimes dynamic algorithms) is concerned with updating a solution relative to a certain graph after some modifications have been made. This field may provide interesting concepts and results, such as the fact that, for NP-hard problems, knowing the optimal solution for a certain graph does not help solving another graph, even if their symmetric distance is small. Other concepts of interest include *sensitivity analysis* in linear programming, that computes the extent of changes to which a solution remains optimal. All these techniques are relevant for some types of temporal graph problems, where a solution to the current snapshot must be updated for the next snapshot.

Similarly, the field of online algorithms could provide interesting tools for tackling problems on *infinite* temporal graphs (see also Theme T2), whether the property of interest relates to the current snapshot or is itself temporal. Online techniques may also be appropriate for finite problems. For instance, the algorithm in [50] for computing non dominated persistent connected components is an online algorithm, and in fact, no offline algorithm of better complexity exists. The same holds for [22]. Note that, in the former, partial solutions computed at time  $t$  are solutions of the problem for time interval  $[0, t]$ , i.e. the solutions are prefix-stable, which is not the case for all problems.

We will also review, in the framework of this theme, a number of simple techniques such as the ones mentioned in Theme C1, including basic reductions from and to static problems defined on the union and/or intersection of the snapshots, or problems reducible to static questions about the time expanded graph (when looking for temporal paths or flows).

Some generic techniques have recently been developed for problems defined on a sliding window, in the context of coloring and matching. In [43], a vertex coloring must verify the classical condition at least once in every time window of size  $\Delta$ . The authors show that this problem admits a polynomial kernel in the number of vertices (for a fixed number of colors). Kernel techniques, in relation to fixed-parameter tractability (see also Theme T1) may offer interesting ground for developing generic techniques for various window based problems (and beyond). Similarly, [8] considers kernels for the matching problem in sliding windows. Interestingly, they show that the problem can be transformed into a static problem instance whose size is independent of the graph lifetime (but exponential in the size of the input). Such transformations are reminiscent of time expanded graphs, although they are not based on the same ideas.

Another direction, perhaps more speculative, will focus on brand new algorithmic techniques. This will probably be related also to the design of advanced data structures (Theme A1). It is expected that a number of these techniques address problems about temporal reachability, where the order of events matters and which cannot therefore be simply defined in terms of union or intersection of snapshots. We will investigate the similarity between existing temporal algorithms for identifying common techniques. For example, the *profile* problem consists in computing, for a given source, the arrival time to a destination as a function of the departure time from the source.

It appears that some algorithms for this problem are similar to the best algorithms for finding a temporal path with shortest duration (i.e. a fastest temporal path). This similarity was observed in [13], and used to design a generic algorithm allowing to find minimum-cost temporal paths for a large variety of costs such as fewest number of edges or minimum travel time (shortest) when different algorithms were proposed for the large variety of classical costs. We will investigate in particular algorithms for finding specific substructures.

Finally, we will investigate possible approaches for decomposing a temporal graph as a general technique for reducing the size of a problem. We will in particular investigate how to extend the notion of module, which is at the core of the modular decomposition. Several such extensions are possible for hypergraphs and all result in polynomial time decomposition algorithms [35]. We will investigate decompositions associated to other graph combinations such as direct products that play a special role in median graphs. A temporal version could be adapted to temporal graphs where each snapshot is a matching (also called *simple* temporal graphs in [20]).

**Objectives:**

1. Make a survey of techniques from related domains (e.g. from re-optimization).
2. Unify existing techniques for temporal graph problems.
3. Design new techniques of ubiquitous use in temporal graphs.

**Related themes:** C1,C2,C3,T1,T3,A1

## 1.6 Risk management and methodology

Extending graph theory and algorithms to temporal graphs appears as a challenging program as various options are possible for generalizing a given graph concept to the temporal setting. The originality of our approach consists in trying to classify how static problems are extended to the temporal setting, and to focus on those that are intrinsically temporal. We think that this approach is rather safe as we already have a simple criterion for such identification through the sensibility to reshuffling of snapshots, even though we will also consider other such criteria. It is also safe to restrict our research space. The associated risk is that we will probably focus on the most difficult versions of the problems: those where new algorithmic techniques are the most needed.

Our main methodology for overcoming such difficulties is to propose a large variety of approaches for understanding the boundaries where interesting problems become difficult. In particular, we review several classes of possible assumptions that can help with respect to tractability. The themes and objectives we propose are based both on our experience of temporal graphs and on our expertise with the classical properties bringing tractability in static graphs. We have a large list of objectives around tractability and algorithm design, and reaching a significant fraction of them would already be a success.

We now detail the risk estimated for all themes. For each theme, we listed several objectives, usually mixing both ambitious ones and more restricted ones where we are rather confident of obtaining results. This is the case for Themes C2, C3, C4, T2, A1, A2, for which the frontier between low and higher risk objectives is clear. As already said, the risks in the objectives listed in C1 are rather low. In Theme T1, the risks on restricting snapshots or the underlying graph seem moderate from preliminary works, while we are conscious that the part about temporal parameters has higher risk. However, we can rely on the growing knowledge corpus from parameterized complexity. For instance, graph parameters are usually related to graph classes and we can first restrict the study to the most basic temporal graph classes. Theme T3 about forbidden temporal patterns is also quite ambitious, but we think that we have sufficiently many directions for exploring it. Theme A3 which aims at a breakthrough in the understanding of temporal graph

algorithms is certainly the most ambitious. That is why the general organization of the project aims at providing various distinct tracks towards this goal.

Even if the project proposes ten different themes and many objectives, they are tightly related to each other. For each axis, we already enumerated the related themes. We now outline some of these links and the timeline we envision. Theme *C1* that addresses the fundamental question of what makes a problem intrinsically temporal will naturally enhance the starting ideas of most of the other themes, and activities in this area should start as early as possible. It is expected the teams will be able to provide some clear answers before the end of the third year. In the same way, the other themes related to classification questions (*C2, C3, C4*) should start from the beginning, and hopefully will provide main results during the first half of the project. On the other hand, some themes will highly benefit from the reflections led in these themes, hence they are not scheduled from the beginning of the project. Obviously, the first reflections about generation problems in *C4* will boost the search for efficient generation algorithms in *A2*. In the same manner, *T3* will rely on *C2*: Among the substructures identified as the core of a given problem, which ones are identified as cornerstones for tractability? Theme *C2* should also help building new temporal graph class hierarchies (theme *T2*). Finally, theme *A3* which will provide new insight in the design of algorithms specific for temporal graph problems will benefit from the results of most of the themes, especially from *C2, C3, T1, T3*, and *A1*. Experimenting on these algorithms calls for the design of adhoc instances provided by *A2*.

The following diagram provides an overview of the project planning. The first line of the diagram represents the global project management, and the deliveries: four annual reports, plus the final report. Each line represents the time span of one theme. Obviously, these spans are only indicative as each theme is not seen as a task with easy to specify deliverables and milestones, but as a research question that we will, at least partially, answer. Note also in the first column, the involvement of the three partners. The participation of each partner to most of the themes (see also the table at the beginning of the document for individual involvement) guarantees a real cooperation between the three partners, even if for management motives, each axis will be coordinated by one different partner.

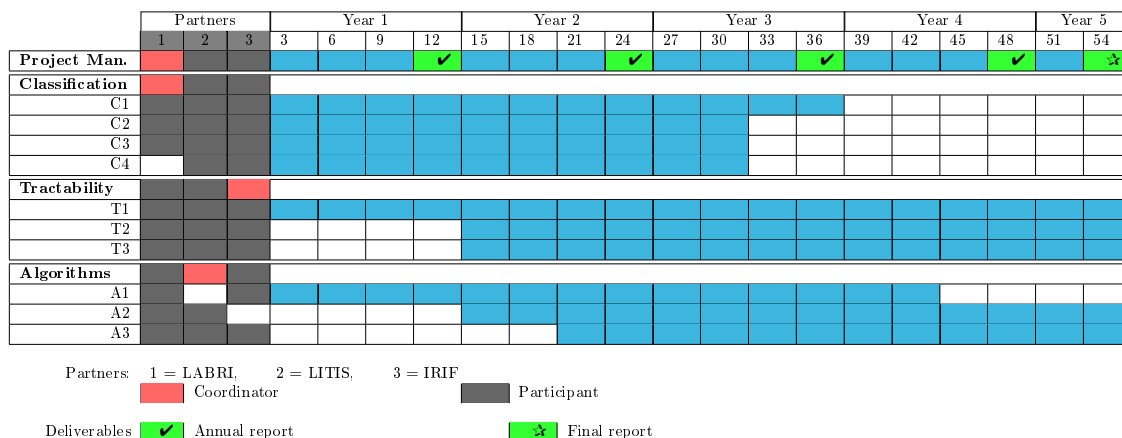


Figure 6: Task Chronograph

## 2 Project organization and means implemented

### 2.1 Coordinator

Arnaud Casteigts is an associate professor at University of Bordeaux. He is the head of the distributed algorithm group at LaBRI. His research interests are in theoretical computer science, with

a focus on algorithms (distributed and non-distributed) and dynamic networks. He is associate editor of the TCS journal. He was the local scientific coordinator of a previous ANR project (ESTATE: Autonomic Computing in distributed and highly dynamic systems and networks, 2016). He has co-supervised three PhD theses on temporal graphs, co-organized several workshops on this topic (DGDC 2016, CoDyn 2017, and the first Dagstuhl seminar on temporal graphs in 2021). He is also involved in the creation of the new born SAND conference, which aims at becoming the main international venue on dynamic networks. He is internationally renowned as one of the main contributors of temporal graph theory.

## 2.2 Consortium

- The LaBRI partner has a strong background in graph theory and distributed algorithms. Its members have been involved in the early development of temporal graph theory, through the lenses of highly-dynamic communication networks, self-stabilization, mobile agents, and transportation mobility (four PhDs on these topics in the past 10 years, and the ESTATE ANR project mentioned above). The team is composed of Arnaud Casteigts (global coordinator, UB: University of Bordeaux), Cyril Gavoille (UB), Nicolas Hanusse (CNRS), Colette Johnen (UB), and Ralf Klasing (CNRS).
- The LITIS partner has been considering temporal graphs since the first version of the Graph-Stream software [27]. More recently they have been involved in the design of optimisation algorithms and in the study and design of generation processes (through three PhDs, two postdocs and one regional project). The team is composed of Eric Sanlaville (scientific coordinator), Stefan Balev, Frédéric Guinand, Yoann Pigné, and Mathilde Vernet, now assistant professor at Avignon University.
- The IRIF partner brings expertise in structural aspects of graphs, both with theoretical perspectives (excluded minor, colorability, reconfiguration) and practical tractability (skeleton dimension, diameter certificates). The team has a particular focus on practical distance labelings and has experience in routing algorithms for temporal graphs in the context of transportation networks (one PhD, collaborations with Instant-System and D-ICE Engineering). It is composed of Laurent Viennot (scientific coordinator, INRIA), Michel Habib (emeritus professor), Fabien de Montgolfier (UP: University of Paris Cité), Reza Naserasr (CNRS), Mikael Rabie (UP).

The three partners of this project have been involved in several works addressing (presumably intrinsic) temporal problems, ranging over the three axes: classification, tractability, algorithms, and considering structural, generation and optimisation temporal problems. Some of these works have been quite influential in the domain. The table below only mentions the projects where the coordinators are implied at the starting time of TEMPOGRAL. The synergy between the teams will be ensured by frequent meetings and visits, and long stays of the people recruited for the project, namely one PhD student at LaBRI and one at LITIS, and two post-docs at IRIF. In particular, the PhD student in Le Havre (resp. in Bordeaux) will spend 6 months in Bordeaux (resp. Le Havre), and one postdoc at IRIF will also spend 6 months in Bordeaux.

Name	Involv. (months)	Funding agency and amount	Project title	Coordinator	Dates
Casteigts	5	ANR 401 k€	Dreamy	T. Nowak & M. Fuegger, LISN & LMF	2021- 2025
Sanlaville	8	Normandy Region, 228 k€	DYNAMIC NETWORKS	P. Dorbec, GREYC	2021- 2022
Viennot	24	ANR 701 k€	Multimod	D. Coudert Inria	2018- 2023
Viennot	6	DGA 270 k€	Brainside	E. Natale Inria	2019- 2023

## 2.3 Justification of requested resources

The grants requested by our consortium are of three types: (i) *Staff*, mainly two PhD students and two years of Postdoc, plus five Master internships; (ii) *Equipments* laptops, books and little devices dedicated to provide decent resources to the students and participants, and (iii) *Travel costs*, mainly used to visit each others and to disseminate our results at international events.

More specifically, one meeting will be organized each year between all members of the project (hence a total of five meetings). Four two-week visits are budgeted for each partner, so that members of one team could spend some time in the other partner teams, thus supporting real collaborative work. It is also expected that PhD students and post-docs financed by the project will spend several months with another team of the project, to promote a deep cooperation. Finally, a significant part of the travel budget is allowed for knowledge dissemination at national and international levels, see below.

The PhD student recruited by the LITIS will work primarily on the search for specific algorithms to solve temporal optimization problems. Hence, this student will contribute to themes *C3* and *A3*, and participate in the thinking around theme *C1*. It is also expected that the student will follow closely the works about tractability. The PhD student at LaBRI will mainly work on themes related to tractability, in particular on fixed-parameter tractability (T1) and hierarchies of temporal graph classes (T2). He or she will apply this general approach (of restricting the instances of a problem) to the search for temporal substructures (theme C2) and develop new techniques to exploit the restrictions (A3). IRIF will host two postdocs on advanced data structures (theme A1, in relation to C2, T1, and T2) and on problems with restricted evolution of the solution (theme C1 in relation with T1, T2, and A3).

The amounts in euros are summarized in the following table:

Type	Description	LABRI	LITIS	IRIF	ALL
Staff	2 PhD scholarships	117 000	110 052		227 052
	2 years of Postdoc			101 520	101 520
	Master Internships	4 920	4 800	3 606	13 326
	<i>Subtotal</i>	121 920	114 852	105 126	<b>341 898</b>
Equipments	PCs and Laptops	4 000	4 000	4 000	12 000
	Books and little devices	1 000	1 000	1 000	3 000
	<i>Subtotal</i>	5 000	5 000	5 000	<b>15 000</b>
Travel costs	Project Meetings	10 500	7 500	9 000	27 000
	Visits	6 000	6 000	6 000	18 000
	International Conferences	15 000	15 000	15 000	45 000
	National Conferences	8 000	8 000	8 000	24 000
	<i>Subtotal</i>	39 500	36 500	38 000	<b>112 400</b>
<b>Total</b>		166 420	156 352	148 126	<b>470 898</b>
+Env.Costs 13%		188 054	176 677	167 382	532 114

## 3 Impact and benefits of the project

### 3.1 Scientific, economic, social impact

Graphs have been a fundamental modeling tool in science since the second world war at least. But as many real-world phenomena are dynamic, they fall beyond the expressivity of standard graphs. This project is launched on the belief that temporal graphs will prove to be a major modeling tool for many application fields. Indeed, they have already been used with significant success these two last decades in statistical physics, communication networks, distributed algorithms, logistics, and social networks. These researches are motivated by the high impact of these different domains in the economic and social spheres. Consider for instance the logistic challenges emphasized by recent world crises, or the large demand for a better understanding of the mechanisms that rule the growth and functioning of social networks.



However, these early studies of temporal graphs have been led independently, which resulted in the difficulty to unify concepts and vocabulary, hence to slow the scientific advances. The scientific community recently proposed some unification initiatives, like multi-disciplinary conferences on complex networks, or the new SAND conference (Symposium on Algorithmic Foundations of Dynamic Networks) that focuses on fundamental aspects of dynamic networks. The TEMPOGRAL project proposes to consider these fundamental aspects independently of the application fields, with the objective to provide new insights and results which, in turn, should allow significant advances in the different application fields of temporal graphs. The large experience and skills of the three partners, and their complementarity, will allow to reach these objectives and to publish the obtained results in the best international journals and conferences. Finally, the project will contribute to organize the French community on temporal graphs, and therefore to increase its international visibility.

TEMPOGRAL will participate in the training of at least 7 students (2 PhD and 5 Master students). This represents roughly two years in internships and six years for PhD students. They will gain strong skills in the area and will be encouraged to go on research activities or to take up positions in the research department of companies related to modeling and computing after they finish their university training.

### 3.2 Dissemination strategy

Research results from TEMPOGRAL will be presented at the leading international conferences and in the leading scientific journals concerned by graph theory, to generate a level of awareness and constructive feedback from the scientific community. At the national level, the new results on temporal graphs will be presented to the community through the working group on graphs of the CNRS GdR IM (Computer science and Mathematics).

TEMPOGRAL members will also contribute to conferences and workshops that seek to attract members from the neighbor communities concerned by temporal graphs, like complex networking (via Complex Network international conference), distributed computing, operations research. Joint meetings may be organized with ANR DUCAT that uses temporal graphs as models of faults in distributed systems. Rapprochements with the working axis CAGDO: Complexity Approximation and Graphs for Decision and Optimization, of CNRS GdR RO (Operations Research), and the Distributed System cluster of CNRS GdR RSD (Networks and distributed systems) are also planned.

Information dissemination will also be carried out through the events organized by TEMPOGRAL members, like seminars, special sessions, workshops. Organizing a summer school centered on temporal graphs, concepts and usage, is also an objective of TEMPOGRAL.

### 3.3 Success Criteria

We will use classical evaluation criteria in academic research, namely: *(i)* the number of scientific publications in top-level conferences and journals, *(ii)* the number of scientific publications in communities which are not the original community of the authors (to show the success of the association of the complementary teams, and of the dissemination policy toward application fields of temporal graphs), *(iii)* the number of effective collaborations with other projects, like ANR projects, and *(iv)* the organization of open workshops and seminars to promote the results.

## 4 Bibliography

- [1] Takuya Akiba, Yoichi Iwata, and Yuichi Yoshida. Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling. In *WWW*, pages 237–248. ACM, 2014.
- [2] Eleni C Akrida, George B Mertzios, Paul G Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107:108–123, 2020.

- [3] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.
- [4] Karine Altisen, Stéphane Devismes, Anaïs Durand, **Colette Johnen**, and Franck Petit. On implementing stabilizing leader election with weak assumptions on network dynamics. In *Proc of ACM Symposium on Principles of Distributed Computing (PODC)*, pages 21–31, 2021.
- [5] Chen Avin, Zvi Lotker, Yinon Nahum, and David Peleg. Improved degree bounds and full spectrum power laws in preferential attachment networks. In *KDD*, pages 45–53. ACM, 2017.
- [6] Kyriakos Axiotis and Dimitris Fotakis. On the size and the approximability of minimum temporally connected subgraphs. *arXiv preprint arXiv:1602.06411*, 2016.
- [7] Hannah Bast, Daniel Delling, Andrew V. Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. Route planning in transportation networks. *CoRR*, abs/1504.05140, 2015.
- [8] Julien Baste, Binh-Minh Bui-Xuan, and Antoine Roux. Temporal matching. *Theor. Comput. Sci.*, 806:184–196, 2020.
- [9] Olivier Beaumont, Tobias Castanet, **Nicolas Hanusse**, and Corentin Travers. Approximation algorithm for estimating distances in distributed virtual environments. In *26th Int. Conf. on Parallel and Distributed Computing (EUROPAR)*, volume 12247 of *LNCS*, pages 359–375. Springer, 2020.
- [10] Matthias Bentert, Anne-Sophie Himmel, André Nichterlein, and Rolf Niedermeier. Efficient computation of optimal temporal walks under waiting-time constraints. *Appl. Netw. Sci.*, 5(1):73, 2020.
- [11] S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *Proceedings of the 2nd International Conference on Ad Hoc, Mobile and Wireless Networks (AdHoc-Now)*, pages 259–270, Montreal, Canada, 2003. Springer.
- [12] Marthe Bonamy, Paul Ouvrard, **Mikaël Rabie**, Jukka Suomela, and Jara Uitto. Distributed recoloring. In *DISC*, volume 121 of *LIPICs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [13] Filippo Brunelli, Pierluigi Crescenzi, and **Laurent Viennot**. On computing pareto optimal paths in weighted time-dependent networks. *Information Processing Letters*, 168:106086, 2021.
- [14] B. Bui-Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2):267–285, April 2003.
- [15] **Arnaud Casteigts**. *A Journey Through Dynamic Networks (with Excursions)*. Habilitation à diriger des recherches, Université de Bordeaux, June 2018.
- [16] **Arnaud Casteigts**, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *Int. J. Parallel Emergent Distributed Syst.*, 27(5):387–408, 2012.
- [17] **Arnaud Casteigts**, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. In *31st Int. Symp. on Algorithms and Computation (ISAAC)*, 2020.
- [18] **Arnaud Casteigts**, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, pages 1–49, 2021.
- [19] **Arnaud Casteigts**, Joseph G. Peters, and Jason Schoeters. Temporal cliques admit sparse spanners. In *46th Int. Coll. on Automata, Languages, and Programming (ICALP)*, volume 132 of *LIPICs*, pages 129:1–129:14, 2019.
- [20] **Arnaud Casteigts**, Joseph G. Peters, and Jason Schoeters. Temporal cliques admit sparse spanners. *Journal of Computer and System Sciences*, 121:1–17, 2021.
- [21] **Arnaud Casteigts**, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp thresholds in random simple temporal graphs. In *62nd Symp. on Foundations of Computer Science (FOCS)*, 2021.
- [22] **Arnaud Casteigts**, **Ralf Klasing**, Yessin Neggaz, and Joseph Peters. Computing parameters of sequence-based dynamic graphs. *Theory of Computing Systems*, 63:394–417, 2019.
- [23] Keren Censor-Hillel and **Mikaël Rabie**. Distributed reconfiguration of maximal independent sets. *Journal of Computer and System Sciences*, 112:85–96, 2020.
- [24] A. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri. Flooding time in edgemarkovian dynamic graphs. In *Proceedings of the 27th ACM Symposium on Principles of distributed computing (PODC)*, pages 213–222, Toronto, Canada, 2008. ACM.
- [25] Daniel Delling, Julian Dibbelt, Thomas Pajor, and Renato F. Werneck. Public transit labeling. In *SEA*, volume 9125 of *Lecture Notes in Computer Science*, pages 273–285. Springer,

- 2015.
- [26] Vida Dujmović, Louis Esperet, **Cyril Gavoille**, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). In *61th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2020.
  - [27] Antoine Dutot, **Frédéric Guinand**, Damien Olivier, and **Yoann Pigné**. Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. In *Emergent Properties in Natural and Artificial Complex Systems (in conj with ECCS)*, 2007.
  - [28] Josep Díaz, Dieter Mitsche, and Xavier Pérez-Giménez. On the connectivity of dynamic random geometric graphs. pages 601–610, 01 2008.
  - [29] Jessica Enright, Kitty Meeks, and Fiona Skerman. Assigning times to minimise reachability in temporal graphs. *Journal of Computer and System Sciences*, 115:169–186, 2021.
  - [30] Laurent Feuilloley and **Michel Habib**. Graph classes and forbidden patterns on three vertices. *SIAM J. Discret. Math.*, 35(1):55–90, 2021.
  - [31] Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020.
  - [32] Laetitia Gauvin, Mathieu Génois, Márton Karsai, Mikko Kivelä, Taro Takaguchi, Eugenio Valdano, and Christian L. Vestergaard. Randomized reference models for temporal networks. *ArXiv*, abs/1806.04032, 2018.
  - [33] Emmanuel Godard and Dorian Mazaauric. Computing the dynamic diameter of non-deterministic dynamic networks is hard. In *10th International Symposium on Algorithms for Sensor Systems (ALGOSENSORS)*, pages 88–102, 2014.
  - [34] Leonidas J. Guibas. Kinetic data structures. In Dinesh P. Mehta and Sartaj Sahni, editors, *Handbook of Data Structures and Applications*. Chapman and Hall/CRC, 2004.
  - [35] **Michel Habib**, **Fabien de Montgolfier**, Lalla Mouatadid, and Mengchuan Zou. A general algorithmic scheme for modular decompositions of hypergraphs and applications. In *IWOCA*, volume 11638 of *Lecture Notes in Computer Science*, pages 251–264. Springer, 2019.
  - [36] Petter Holme. Modern temporal network theory: A colloquium. *CoRR*, abs/1508.01303, 2015.
  - [37] David Ilcinkas, **Ralf Klasing**, and Ahmed Mouhamadou Wade. Exploration of constantly connected dynamic graphs based on cactuses. In *International Colloquium on Structural Information and Communication Complexity*, pages 250–262. Springer, 2014.
  - [38] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. In *Proceedings of 32nd ACM Symposium on Theory of Computing (STOC)*, pages 504–513, Portland, USA, 2000. ACM.
  - [39] David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820 – 842, 2002.
  - [40] Adrian Kosowski and **Laurent Viennot**. Beyond highway dimension: small distance labels using tree skeletons. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1462–1478. SIAM, 2017.
  - [41] Adrian Kosowski, Przemyslaw Uznanski, and **Laurent Viennot**. Hardness of exact distance queries in sparse graphs through hub labeling. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 272–279, 2019.
  - [42] Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):1–29, 2018.
  - [43] George B. Mertzios, Hendrik Molter, and Viktor Zamaraev. Sliding window temporal graph coloring. *CoRR*, abs/1811.04753, 2018.
  - [44] Othon Michail and Paul G Spirakis. Traveling salesman problems in temporal graphs. *Theoretical Computer Science*, 634:1–23, 2016.
  - [45] Kamaldeep Singh Oberoi and Géraldine Del Mondo. Graph-based Pattern Detection in Spatio-Temporal Phenomena. In *16th Spatial Analysis and Geomatics Conference (SAGEO 2021)*, La Rochelle (virtuel), France, May 2021.
  - [46] Jason Schoeters. *Contributions to temporal graph theory and mobility-related problems*. PhD thesis, Bordeaux, 2021.
  - [47] Manuel Sorge and Mathias Weller et al. The graph parameter hierarchy, 2018. 2020.
  - [48] Michele Tirico, **Stefan Balev**, Antoine Dutot, and Damien Olivier. Morphogenesis of complex networks: a reaction diffusion framework for spatial graphs. In *International Conference on Complex Networks and their Applications*, pages 769–781. Springer, 2018.

- [49] **Mathilde Vernet**, Maciej Drozdowski, **Yoann Pigné**, and **Eric Sanlaville**. A theoretical and experimental study of a new algorithm for minimum cost flow in dynamic graphs. *Discrete Applied Mathematics*, 296:203–216, 2021.
- [50] **Mathilde Vernet**, **Yoann Pigné**, and **Eric Sanlaville**. A study of connectivity on dynamic graphs: Computing persistent connected components. *I'OR: a Quarterly Journal of Operations Research*. accepted march 2022.
- [51] Christian L Vestergaard, Mathieu Géniois, and Alain Barrat. How memory generates heterogeneous dynamics in temporal networks. *Physical Review E*, 90(4):042805, 2014.