Simple, strict, proper, happy: A study of reachability in temporal graphs*

Arnaud Casteigts[†], Timothée Corsini[‡], Writika Sarkar[§]

Abstract

Dynamic networks are a complex subject. Not only do they inherit the complexity of static networks (as a particular case); they are also sensitive to definitional subtleties that are a frequent source of confusion and incomparability of results in the literature.

In this paper, we take a step back and examine three such aspects in more details, exploring their impact in a systematic way; namely, whether the temporal paths are required to be *strict* (i.e., the times along a path must increasing, not just be non-decreasing), whether the time labeling is *proper* (two adjacent edges cannot be present at the same time) and whether the time labeling is *simple* (an edge can have only one presence time). In particular, we investigate how different combinations of these features impact the expressivity of the graph in terms of reachability.

Our results imply a hierarchy of expressivity for the resulting settings, shedding light on the loss of generality that one is making when considering either combination. Some settings are more general than expected; in particular, proper temporal graphs turn out to be as expressive as general temporal graphs where non-strict paths are allowed. Also, we show that the simplest setting, that of *happy* temporal graphs (i.e., both proper and simple) remains expressive enough to emulate the reachability of general temporal graphs in a certain (restricted but useful) sense. Furthermore, this setting is advocated as a target of choice for proving negative results. We illustrates this by strengthening two known results to happy graphs (namely, the inexistence of sparse spanners, and the hardness of computing temporal components). Overall, we hope that this article can be seen as a guide for choosing between different settings of temporal graphs, while being aware of the way these choices affect generality.

Keywords: Temporal graphs; Temporal reachability; Reachability graph; Expressivity.

1 Introduction

In the context of this paper, a temporal graph is a labeled graph $\mathcal{G} = (V, E, \lambda)$ where V is a finite set of vertices, $E \subseteq V \times V$ a set of undirected edges, and $\lambda : E \to 2^{\mathbb{N}}$ a function assigning at least one time label to every edge, interpreted as presence times. These graphs can model various phenomena, ranging from dynamic networks – networks whose structure changes over the time – to dynamic interactions over static (or dynamic) networks. These graphs have found applications in biology, transportation, social networks,

*This work has been partially funded by the ANR project TEMPOGRAL (ANR-22-CE48-0001).

[†]University of Geneva, Switzerland.

[‡]LaBRI, CNRS, Univ. Bordeaux, Bordeaux INP, France.

[§]Chennai Mathematical Institute, India

robotics, scheduling, distributed computing, and self-stabilization, to name a few. Although more complex formalisms have been defined and extensively studied (see e.g. [14] or [30]), several features of temporal graphs remain not well understood even in the most restricted settings.

A fundamental aspect of temporal graphs is reachability, commonly characterized in terms of the existence of temporal paths; i.e., path which traverses edges in chronological order. There has been a large number of studies related to temporal reachability in the past two decades, seen from various perspectives, e.g. k-connectivity and separators [28, 25, 20], components [7, 4, 2, 31, **?**, **?**], feasibility of distributed tasks [14, 26, 3, 9], schedule design [11], data structures [12, 33, 31, 10], reachability minimization [22], reachability with additional constraints [12, 15], temporal spanners [4, 2, 17, 8], path enumeration [23], random graphs [6, 18], exploration [27, 21, 24], cops and robbers [**?**, **?**], and temporal flows [1, 32], to name a few (many more exist). Over the course of these studies, it has become clear that temporal connectivity differs significantly from classical reachability in static graphs. To start with, it is not transitive, which implies that two temporal paths (also called journeys) are not, in general, composable, and consequently, connected components do not form equivalence classes. This explains, in part, why many tractable problems in static graphs become hard when transposed to temporal graphs. Further complications arise, such as the conceptual impact of having an edge appearing multiple times, and that of having adjacent edges appearing at the same time. These aspects, while innocent-looking, have a deep impact on the answers to many structural and algorithmic questions.

In this paper, we take a step back, and examine the impact of such aspects; in particular *strictness* (should the times along a path increase or only be non-decreasing?), *properness* (can two adjacent edges appear at the same time?) and *simpleness* (do the edges appear only once or several times?). We look at the impact of these aspects from the point of view of temporal reachability, and more precisely, how they restrict it. The central tool is the notion of reachability graph, defined as the static directed graph where an arc exists if and only if a temporal path exists in the original temporal graph.[1] It turns out that the above aspects have a strong impact on the kind of reachability graph one can obtain from a temporal graph. Precisely, we establish four separations between various combinations (called *settings*) of the above parameters. On the other hand, we also present three reachability-preserving transformation between settings, which show that certain settings are at least as expressive as others.

By combining the separations and transformations together with arguments of containment, we obtain an almost complete hierarchy of expressivity of these settings in terms of reachability. This hierarchy clarifies the extent to which the choice of a particular setting impacts generality, and as such, can be used as a guide for future research in temporal graphs. Indeed, the above three aspects (strictness, properness, simpleness) are a frequent source of confusion and of incomparability of results in the literature. Furthermore, many basic questions remain unresolved even in the most restricted setting. For this reason, and somewhat paradoxically, we advocate the study of the simplest model, that of *happy* temporal graphs (i.e., both proper and simple), where all the above subtleties vanish. Another reason is that, despite being the least expressive setting, happy graphs remain general enough to capture *certain features* of general temporal reachability. Finally, negative results in this setting are *de facto* stronger than in all the other settings. In guise of illustration, we strengthen two existing negative results to the happy setting. Namely, finding temporal components of a

[1] This concept was called the *transitive closure of journeys* in [7, 10, 16]; we now avoid this term because reachability is not transitive, which makes it somewhat misleading.

given size remains difficult even in happy graphs and the existence of $o(n^2)$-sparse temporal spanners is also not guaranteed even in happy graphs. Both results were initially obtained in more general settings (respectively, in the non-proper, non-simple, non-strict setting [7] for the former, and in the non-proper, simple, non-strict setting [4] for the latter).

The paper is organized as follows. In Section 2, we give some definitions and argue that the above aspects deserve to be studied for their own sake. In Section 3, we present the four separations and the three transformations, together with the resulting hierarchy. In Section 4, we strengthen the hardness of temporal components and the counter-example for sparse spanners to the setting of happy graphs, and motivate their study further. Finally, we conclude in Section 5 with some remarks.

2 Temporal Graphs

Given a temporal graph $\mathcal{G} = (V, E, \lambda)$, the static graph $G = (V, E)$ is called the *footprint* of $\mathcal{G}$. Similarly, the static graph $G_t = (V, E_t)$ where $E_t = \{e \in E \mid t \in \lambda(e)\}$ is the *snapshot* of $\mathcal{G}$ at time t. A pair (e, t) such that $e \in E$ and $t \in \lambda(e)$ is a *contact* (or temporal edge). The range of λ is called the *lifetime* of $\mathcal{G}$, of length τ. A *temporal path* (or journey) is a sequence of contacts $\langle(e_i, t_i)\rangle$ such that $\langle e_i\rangle$ is a path in the footprint and $\langle t_i\rangle$ is non-decreasing.

The reachability relation based on temporal paths can be captured by a *reachability graph*, i.e. a static directed graph $\mathcal{R}(\mathcal{G}) = (V, E_c)$, such that $(u, v) \in E_c$ if and only if a temporal path exists from u to v. A graph $\mathcal{G}$ is *temporally connected* if all the vertices can reach each other at least once (i.e., $\mathcal{R}(\mathcal{G})$ is a complete directed graph). The class of temporally connected graphs (TC) is arguably one of the most basic classes of temporal graphs, along with its infinite lifetime analog TC^R, where temporal connectivity is achieved infinitely often (i.e., recurrently).

In what follows, we drop the adjective "temporal" whenever it is clear from the context that the considered graph (or property) is temporal.

2.1 Strictness / Properness / Simpleness

The above definitions can be restricted in various ways. In particular, one can identify three restrictions that are common in the literature, although they are sometimes considered implicitly and under various names:

- *Strictness*: A temporal path $\langle(e_i, t_i)\rangle$ is *strict* if $\langle t_i\rangle$ is increasing.
- *Properness*: A temporal graph is *proper* if $\lambda(e) \cap \lambda(e') = \emptyset$ whenever e and e' are incident to a same vertex (i.e., λ is locally-injective).
- *Simpleness*: A temporal graph is *simple* if λ is single-valued; that is, every edge has a single presence time.

Strictness is perhaps the easiest way of accounting for traversal time for the edges. Without such restriction (i.e., in the default *non-strict* setting), a journey can traverse arbitrarily many edges at the same time step. The notion of *properness* is related to the one of strictness, although not equivalent. Properness forces all the journeys to be strict, because adjacent edges always have different time labels. However, if the graph is non-proper, then considering strict or non-strict journeys does have an impact, thus distinguishing both concepts is important. We will call *happy* a graph that is both proper and simple, for reasons that will become clear later.

Application-wise, proper temporal graphs arise naturally when the graph represent mutually exclusive interactions. Proper graphs also have the advantage that λ induces a proper coloring of the contacts (interpreting the labels as colors). Finally, *simpleness* naturally accounts for scenarios where the entities interact only one time. It is somewhat unlikely that a real-world system has this property; however, this restriction has been extensively considered in well studied subjects (e.g. in gossip theory).

Note that simpleness and properness are properties of the *graph*, whereas strictness is a property of the *temporal paths* in that graph. Therefore, one may either consider a strict or a non-strict setting in a same temporal graph. The three notions (of strictness, simpleness, and properness) interact in subtle ways, these interactions being a frequent source of confusion and incomparability among results. Before focusing on these interactions, let us make a list of the possible combinations. The naive cartesian product of these restrictions leads to eight combinations. However, not all of them are meaningful, since properness removes the distinction between strict and non-strict journeys. Overall, there are six meaningful combinations, illustrated in Figure 1.

- Non-proper, non-simple, strict (1)
- Non-proper, non-simple, non-strict (2)
- Non-proper, simple, strict (4)
- Non-proper, simple, non-strict (5)
- Proper, non-simple (3)
- Proper, simple (= happy) (6)

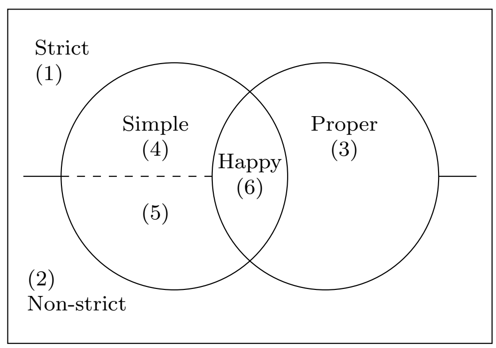


Figure 1: Settings resulting from combining the three properties.

In the name of the settings, "non-proper" refers to the fact that properness is *not required*, not to the fact that it is necessarily not satisfied. In other words, proper graphs are a particular case of non-proper graphs, and likewise, simple graphs are a particular case of non-simple graphs. Thus, whenever non-proper or non-simple graphs are considered, we will omit this information from the name. For instance, setting (1) will be referred to as the (general) strict setting. Finally, observe that strict paths are a special case of non-strict paths, but we do not get an inclusion of the corresponding settings, whose features are actually incomparable (we shall return on that subtle point later).

2.2 Does it really matter? (Example of spanners)

While innocent-looking, the choice for a particular setting may have tremendous impacts on the answers to basic questions. For illustration, consider the spanner problem. Given a graph $\mathcal{G} = (V, E, \lambda)$ such that $\mathcal{G} \in TC$, a *temporal spanner* of $\mathcal{G}$ is a graph $\mathcal{G}' = (V, E', \lambda')$ such that $\mathcal{G}' \in TC$, $E' \subseteq E$, and for all e in E', $\lambda'(e) \subseteq \lambda(e)$. In other words, $\mathcal{G}'$ is a temporally connected spanning subgraph of $\mathcal{G}$. A natural goal is to minimize the size of the spanner, either in terms of number of labels or number of underlying edges. More formally,

Min-Label Spanner
Input: A temporal graph $\mathcal{G}$, an integer k
Output: Does $\mathcal{G}$ admit a temporal spanner with at most k contacts?

Min-Edge Spanner
Input: A temporal graph $\mathcal{G}$, an integer k
Output: Does $\mathcal{G}$ admit a temporal spanner of at most k edges (keeping all their labels)?

The search and optimization versions of these problems can be defined analogously. Unlike spanners in static graphs, the definition does not care about stretch factors, due to the fact that the very existence of small spanners is not guaranteed. In the following, we illustrate the impact of the notions of strictness, simpleness, and properness (and their interactions) on these questions. The impact of strictness is pretty straightforward. Consider the graph $\mathcal{G}_1$ on Figure 2. If non-strict journeys are allowed, then this graph admits $\mathcal{G}_2$ as a spanner (among others), this spanner being optimal for both versions of the problem (3 labels, 3 edges). Otherwise, the minimum spanners are bigger (and different) for both versions: $\mathcal{G}_3$ minimizes the number of labels (4 labels, 4 edges), while $\mathcal{G}_4$ minimizes the number of edges (3 edges, 5 labels). If strictness is combined with non-properness, then there exist a pathological scenario (already identified in [28] and [2]) where the input is a complete temporal graph (see $\mathcal{G}_5$, for example) but none of the edges can be removed without breaking connectivity! Note that $\mathcal{G}_5$ is a simple temporal graph. Simpleness has further consequences. For example, if the input graph is simple and proper, then it cannot

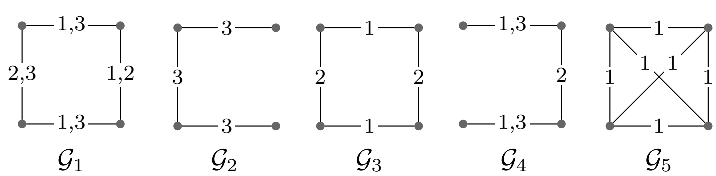


Figure 2: Some temporal graphs on four vertices.

admit a spanning tree (i.e. a spanner of $n-1$ edges) and requires at least $2n-4$ edges (or labels, equivalently, since the graph is simple) [13]. If the input graph is simple and non-proper, then it does not admit a spanning tree if strictness is required, but it does admit one otherwise if and only if at least one of the snapshots is connected (in a classical sense). Finally, none of these affirmations hold in general for non-simple graphs.

If the above discussion seems confusing to the reader, it is not because we obfuscated it. The situation is intrinsically subtle. In particular, one should bear in mind the above subtleties whenever results from different settings are compared with each other. To illustrate such pitfalls, let us relate a recent mistake (fortunately, without consequences) that involved one of the authors. In [4], Axiotis and Fotakis constructed a (non-trivial) infinite family of temporal graphs which do not admit $o(n^2)$-sparse spanners. Their construction is given in the setting of simple temporal graphs, with non-proper labeling and non-strict journeys allowed. The same paper actually uses many constructions formulated in this setting, and a general claim is that these constructions can be adapted to proper graphs (and so strict journeys). Somewhat hastily, the introductions of [18] and [17] claim that the counterexample from [4] holds in happy graphs. The pitfall is that, for some of the constructions in [4], giving up on non-properness (and non-strictness) is only achievable at the cost of using *multiple* labels per edge – a conclusion that we reached in the meantime. To be fair, the authors of [4] never claimed that these adaptations could preserve simpleness, so their claim was actually correct.

Apart from illustrating the inherent subtleties of these notions, the previous observations imply that counter-examples to sparse spanners in happy graphs was in fact still open. In Section 4.2, we show that the spanner construction from [4] can indeed be adaptated to this very restricted setting.

2.3 Happy Temporal Graphs

A temporal graph $\mathcal{G} = (V, E, \lambda)$ is *happy* if it is both proper and simple. These graphs have sometimes been referred to as *simple temporal graphs* (including by the authors), which the present paper now argues is insufficiently precise. Happy graphs are "happy" for a number of reasons. First, the distinction between strict journeys and non-strict journeys can be safely ignored (due to properness), and the distinction between contacts and edges can also be ignored (due to simpleness). Clearly, these restrictions come with a loss of expressivity, but this does not prevent happy graphs from being relevant more generally in the sense that negative results in these graphs carry on to all the other settings. For example, if a problem is computationally hard on happy instances, then it is so in all the other settings. Thus, it seems like a good practice to try to prove negative results for happy graphs first, whenever possible. If this is not possible, then proving it in proper graphs still has the advantage of making it applicable to both strict and non-strict temporal paths alike. Positive results, on the other hand, are not generally transferable; in particular, a hard problem in general temporal graphs could become tractable in happy graphs. This being said, if a certain graph contains a happy subgraph, then whatever pattern can be found in the latter also exists in the former, which enables *some form* of transferability for positive results as well from happy graphs to more general temporal graphs.

In fact, happy graphs coincide with a vast body of literature. Many studies in *gossip theory* and *population protocols* consider the same restrictions, and the so-called *edge-ordered graphs* [19] can also be seen as a particular case of happy graphs where λ is *globally* injective (although the distinction does not matter for reachability). In addition, a number of other existing results in temporal graphs consider such restrictions.

Finally, a nice property of happy graph is that, up to time-distortion that preserve the local ordering of the edges, the number of happy graphs on a certain number of vertices is *finite* – a crucial property for exhaustive search and verification (note that this is also the case of simple graphs, more generally).

We think that the above arguments, together with the fact that many basic questions remain unsolved even in this restricted model, makes happy graphs a compelling class of temporal graphs to be studied in the current state of knowledge.

3 Expressivity of the settings in terms of reachability

As already said, a fundamental aspect of temporal graphs is reachability through temporal paths. There are several ways of characterizing the extent to which two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ have similar reachability. The first three, below, are increasingly more restrictive.

Definition 1 (Reachability equivalence)**.** *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two temporal graphs built on the same set of vertices. $\mathcal{G}_1$ and $\mathcal{G}_2$ are* reachability-equivalent *if $\mathcal{R}(\mathcal{G}_1) \simeq \mathcal{R}(\mathcal{G}_2)$ (i.e. both reachability graphs are isomorphic). By abuse of language, we say that $\mathcal{G}_1$ and $\mathcal{G}_2$ have the "same" reachability graph.*

Definition 2 (Support equivalence)**.** *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two temporal graphs built on the same set of vertices. These graphs are* support-equivalent *if for every journey in either graph, there exists a journey in the other graph whose underlying path goes through the same sequence of vertices.*

Definition 3 (Bijective equivalence)**.** *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two temporal graphs built on a same set of vertices. These graphs are* bijectively equivalent *if there is a bijection σ between the set of journeys of $\mathcal{G}_1$ and that of $\mathcal{G}_2$, and σ is support-preserving.*

The following form of equivalence is weaker.

Definition 4 (Induced reachability equivalence)**.** *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two temporal graphs built on vertices V_1 and V_2, respectively, with $V_1 \subseteq V_2$. $\mathcal{G}_2$ is* induced-reachability equivalent *to $\mathcal{G}_1$ if $\mathcal{R}(\mathcal{G}_2)[V_1] \simeq \mathcal{R}(\mathcal{G}_1)$. In other words, the restriction of $\mathcal{R}(\mathcal{G}_2)$ to the vertices of V_1 is isomorphic to $\mathcal{R}(\mathcal{G}_1)$.*

Observe that bijective equivalence implies support equivalence, which implies reachability equivalence, which implies induced reachability equivalence. Furthermore, support equivalence forces both footprints to be the same (the converse is not true). In this section, we show that some of the settings differ in terms of reachability, whereas others coincide. We first prove a number of *separations*, by showing that there exist temporal graphs in some setting, whose reachability graph cannot be realized in some other settings (Section 3.1). Then, we present three *transformations* which establish various levels of equivalences (Section 3.2). Finally, we infer more relations by combining separations and transformations in Section 3.3, together with further discussions. A complete diagram illustrating all the relations is given in the end of the section (Figure 5 on page 15).

3.1 Separations

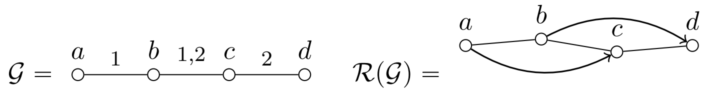
In view of the above discussion, a separation in terms of reachability graphs is pretty general, as it implies a separation for the two stronger forms of equivalences (support-preserving and bijective ones). Before starting, let us state a simple lemma used in several of the subsequent proofs.

Lemma 1. *In the non-strict setting, if two vertices are at distance two in the footprint, then at least one of them can reach the other (i.e. the reachability graph must have at least one arc between these vertices).*

3.1.1 "Simple & strict" *vs.* "strict"

Lemma 2. *There is a graph in the "strict" setting whose reachability graph cannot be obtained from a graph in the "simple & strict" setting.*

Proof. Consider the following non-simple graph $\mathcal{G}$ (left) in a strict setting and the corresponding reachability graph (right). We will prove that a hypothetical simple temporal graph $\mathcal{H}$ with same reachability graph as $\mathcal{G}$ cannot be built in the strict setting. First,



observe that the arc (a, c) in $\mathcal{R}(\mathcal{G})$ exists only in one direction. Thus, a and c cannot be neighbors in $\mathcal{H}$. Since $\mathcal{H}$ is simple and the journeys are strict (and a has no other neighbors in $\mathcal{R}(\mathcal{G})$), the arc (a, c) can only result from the label of ab being strictly less than bc. The same argument holds between bc and cd with respect to the arc (b, d) in $\mathcal{R}(\mathcal{G})$. As a result, the labels of ab, bc, and cd must be strictly increasing, which is impossible since (a, d) does not exist in $\mathcal{R}(\mathcal{G})$. □

As simple graphs are a particular case of non-simple graphs, the following follows.

Corollary 1. *The "simple & strict" setting is strictly less expressive than the "strict" setting in terms of reachability graphs.*

3.1.2 "Non-strict" *vs.* "simple & strict"

Lemma 3. *There is a graph in the "simple & strict" setting whose reachability graph cannot be obtained from a graph in the "non-strict" setting.*

Proof. Consider the following simple temporal graph $\mathcal{G}$ (left) in a strict setting and the corresponding reachability graph (right). Note that a and c are not neighbors in $\mathcal{R}(\mathcal{G})$, due to strictness. For the sake of contradiction, let $\mathcal{H}$ be a temporal graph whose non-strict reachability graph is isomorphic to that of $\mathcal{G}$. First, observe that the footprint of $\mathcal{H}$ must be

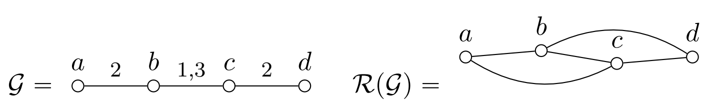
isomorphic to the footprint of $\mathcal{G}$, as otherwise it is either complete or not connected. Call b the vertex of degree two in $\mathcal{H}$. If $\lambda_{\mathcal{H}}(ab) \neq \lambda_{\mathcal{H}}(bc)$, then either a can reach c or c can reach a, and if $\lambda_{\mathcal{H}}(ab) = \lambda_{\mathcal{H}}(bc)$, then both can reach each other through a non-strict journey. In both cases, $\mathcal{R}(\mathcal{H})$ contains more arcs than $\mathcal{R}(\mathcal{G})$. □

As stated in the end of the section, the reverse direction is left open.

3.1.3 "Simple & non-strict" *vs.* "proper"

Lemma 4. *There is a graph in the "proper" setting whose reachability graph cannot be obtained from a graph in the "simple & non-strict" setting.*

Proof. Consider the following proper temporal graph $\mathcal{G}$ (left). Its reachability graph (right) is a graph on four vertices, with an edge between any pair of vertices except a and d (i.e., a diamond). For the sake of contradiction, let $\mathcal{H}$ be a simple temporal graph in the non-strict setting, whose reachability graph is isomorphic to that of $\mathcal{G}$. First, observe that no arcs



exist between a and d in the reachability graph, thus a and d must be at least at distance 3 in the footprint (Lemma 1), which is only possible if the footprint is a graph isomorphic to P_4 (i.e. a path graph on four vertices) with endpoints a and d. Now, let t_1, t_2, and t_3 be the labels of ab, bc, and cd respectively. Since $\{a, b, c\}$ is a clique in the reachability graph (whatever the way identifiers b and c are assigned among the two remaining vertices), they must be temporally connected in $\mathcal{H}$, which forces that $t_1 = t_2$ (otherwise both edges could be travelled in only one direction). Similarly, the fact that $\{b, c, d\}$ is a clique in the reachability graph forces $t_2 = t_3$. As a result, there must be a non-strict journey between a and d, which contradicts the absence of arc between a and d in the reachability graph. □

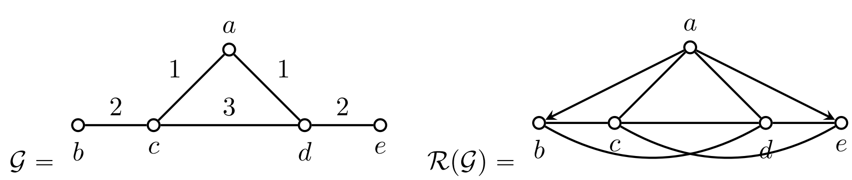
The next corollary follows by inclusion of proper graphs in the non-strict setting.

Corollary 2. *The "simple & non-strict" setting is strictly less expressive than the "non-strict" setting in terms of reachability graphs.*

3.1.4 "simple & proper (i.e. happy)" *vs.* "simple & non-strict"

Lemma 5. *There is a graph in the "simple & non-strict" setting whose reachability graph cannot be obtained in the "happy" setting.*

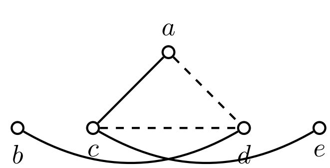
Proof. Consider the following simple temporal graph $\mathcal{G}$ (left) in a non-strict setting and the corresponding reachability graph (right). For the sake of contradiction, let $\mathcal{H}$ be a happy temporal graph whose reachability graph is isomorphic to that of $\mathcal{G}$.



Since a is not isolated in the reachability graph, it has at least one neighbor in $\mathcal{H}$. Vertices b and e cannot be such neighbors, the arc being oneway in the reachability graph, so its neighbors are either c, d, or both c and d. *Wlog*, assume that c is a neighbor (the arguments hold symmetrically for d), we first prove an intermediate statement

Claim 5.1. *The edge bd does not exists in the footprint of $\mathcal{H}$.*

Proof of Claim 5.1 (by contradition). If $bd \in \mathcal{H}$, then $de \notin \mathcal{H}$, as otherwise b and e would be at distance 2 and share at least one arc in the reachability graph (Lemma 1). However, e must have at least one neighbor, thus $ce \in \mathcal{H}$, and by Lemma 1 again $bc \notin \mathcal{H}$. At this point, the footprint of $\mathcal{H}$ must look like the following graph, in which the status of ad and cd is not settled yet.



In fact, ad must exist, as otherwise there is no way of connecting d to a *and* a to d. Also note that the absence of (e, a) in the reachability graph forces $\lambda(ac) < \lambda(ce)$ (remember that $\mathcal{H}$ is both proper and simple), which implies that no journey exists from e to d unless cd is also added to $\mathcal{H}$ with a label $\lambda(cd) > \lambda(ce)$. In the opposite direction, d needs that $\lambda(ad) < \lambda(ac)$ to be able reach e. Now, c needs that $\lambda(cd) < \lambda(bd)$ to reach b. In summary, we must have $\lambda(ad) < \lambda(ac) < \lambda(ce) < \lambda(cd) < \lambda(bd)$, which implies that b cannot reach c. □

By this claim, $bd \notin \mathcal{H}$, thus $bc \in \mathcal{H}$ and consequently $cd \notin \mathcal{H}$ (by Lemma 1). From the absence of (b, a) in the reachability graph, we infer that $\lambda(bc) > \lambda(ac)$. In order for b to reach d, we need that cd exists with label $\lambda(cd) > \lambda(bc)$. To make d to b mutually reachable, there must be an edge ad with time $\lambda(ad) < \lambda(ac)$. Now, the only way for c to reach e is through the edge de, and since there is no arc (e, a), its label must satisfy $\lambda(de) > \lambda(ad)$. Finally, c can reach e (but not through a), so $\lambda(de) > \lambda(cd)$ and c cannot reach e, a contradiction. □

By inclusion of happy graphs in the "simple & non-strict" setting, we have

Corollary 3. *The "simple & proper (i.e. happy)" setting is strictly less expressive than the "simple & non-strict" setting in terms of reachability graphs.*

3.2 Transformations

In this section, we present three transformations. First, we present a transformation from the general non-strict setting to the setting of proper graphs, called the *dilation technique*. Since proper graphs are contained in both the non-strict and strict settings, this transformation implies that the strict setting is at least as expressive as the non-strict setting. This transformation is *support-preserving*, but it suffers from a significant blow-up in the size of the lifetime. Another transformation called the *saturation technique* is presented from the (general) non-strict setting to the (general) strict setting, which is only *reachability-preserving* but preserves the size of the lifetime. Finally, we present an induced-reachability-preserving transformation, called the *semaphore technique*, from the general strict setting to happy graphs. If the original temporal graph is non-strict, one can compose it with one of the first two transformations, implying that *all* temporal graphs can be turned into a happy graph whose reachability graph contains that of the original temporal graph as an induced subgraph. This shows that happy graphs are universal in a weak (in fact, induced) sense.

3.2.1 Dilation technique: "non-strict" → "proper"

Given a temporal graph $\mathcal{G}$ in the non-strict setting, we present a transformation that creates a proper temporal graph $\mathcal{H}$ that is support-equivalent to $\mathcal{G}$ (and thus also reachability-equivalent). We refer to this transformation as the *dilation* technique.

The transformation operates at the level of the snapshots, taken independently, one after the other. It consists of isolating, in turn, every snapshot G_t where some non-strict journeys are possible, and "dilating" it over more time steps in such a way these journeys can be made strict (note that this needs be applied only if G_t contains at least one path of length larger than 1). The subsequent snapshots are shifted in time accordingly. The dilation of a snapshot G_t goes as follows. Without loss of generality, assume that $t = 1$ (otherwise, shift the labels used below by the sum of lifetimes resulting from the dilation of the earlier snapshots). First, we transform G_t into a non-proper temporal graph $\mathcal{G}_t$ whose footprint is G_t itself, and the edges of which are assigned labels $1, 2, ..., k$, where k is the longest path in G_t (with $k \leqslant |V| - 1$). As argued in the proof below, there is a strict journey in $\mathcal{G}_t$ if and only if there is a path (and thus a non-strict journey) in G_t. Now, $\mathcal{G}_t$ can be turned into a proper graph as follows. By Vizing's theorem, the edges of a graph of maximum degree Δ can be properly colored using at most $\Delta + 1$ colors. Let $c : E_t \to [0, \Delta]$ be such a coloring, and let ϵ be a fixed value less than $1/(\Delta + 1)$. Each label of each edge e of E_t is "tilted" in $\mathcal{G}_t$ by a quantity equal to $c(e)\epsilon$. The transformation is illustrated in Figure 3.

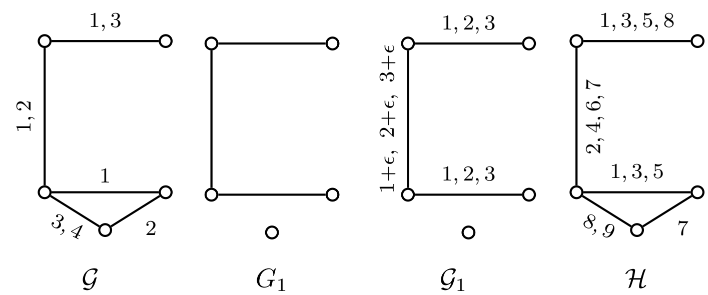


Figure 3: Dilation of the labels of $\mathcal{G}$. Here, a single snapshot, namely G_1 contains paths whose length is larger than 1, so the dilation is only applied to G_1. The transformed snapshot $\mathcal{G}_1$ has 6 labels (instead of 1). It is then recomposed with the other snapshots, whose time labels are shifted accordingly (by 5 time unit), resulting in graph $\mathcal{H}$.

Lemma 6 (Dilation technique)**.** *Given a temporal graph $\mathcal{G}$, the dilation technique transforms $\mathcal{G}$ into a proper temporal graph $\mathcal{H}$ such that there is a non-strict journey in $\mathcal{G}$ if and only if there is a strict journey in $\mathcal{H}$ with same support.*

Proof. (Properness.) Every snapshot is dealt with independently and chronologically, the subsequent snapshots being shifted as needed to occur after the transformed version of the formers, so each snapshot becomes a temporal subgraph of $\mathcal{H}$ whose lifetime occupies a distinct subinterval of the lifetime of $\mathcal{H}$. Moreover, the tilting method based on a proper coloring of the edges guarantees that each of these temporal graphs is proper. Thus, $\mathcal{H}$ is proper.
(Preservation of journeys.) Given a snapshot G_t considered independently, the longest path in G has length $k \leqslant |V| - 1$ and every edge has all the labels from 1 to k, so for every path of length ℓ in G, there is a *strict* journey in this graph, along the same sequence of edges, going over labels $1, 2, ..., \ell$ (up to the tilts, which are all less than 1). Moreover, if a journey exists in $\mathcal{G}_t$, then its underlying path also exists in G_t (since G_t is the footprint of $\mathcal{G}_t$), thus, the dilation of a snapshot is support-preserving. Finally, as all the snapshots occupy a distinct subinterval of the lifetime of $\mathcal{H}$, and the order among snapshots is preserved, the composability of journeys over different snapshots is also unaffected. □

Let us clarify a few additional properties of the transformation. In particular,

Lemma 7. *The running time of the dilation technique is polynomial.*

Proof. For any reasonable representation of $\mathcal{G}$ in memory, one can easily isolate a particular snapshot by filtering the contacts for the corresponding label (if the representation is itself snapshot-based, this step is even more direct). Then, each snapshot has at most $O(n^2)$ edges, so assigning the n required labels to each of them takes at most $O(n^3)$ operations. Using

Misra and Gries coloring algorithm [29], a proper coloring of the edges can be obtained in $O(n^3)$ time per snapshot. Applying the tilt operation takes essentially one operation per label in the transformed snapshot, so $O(n^3)$ again. Finally, these operations must be performed for all snapshots, which incurs a additional global factor of τ, resulting in a total running time of essentially $O(n^3\tau)$. (The exact running time may depend on the actual data structure. It could also be characterized more finely by considering the number of contacts k (temporal edges) as a parameter instead of the rough approximation above, leading to a complexity of $O(kn)$ time steps, e.g. with adjacency lists to describe the snapshots.) ☐

Finally, observe that the dilation technique may incur a significant blow up in the lifetime of the graph. More precisely,

Lemma 8. *Let Δ be the maximum degree of a vertex in any of the snapshot. Then, $\tau_{\mathcal{H}} \leqslant \tau_{\mathcal{G}}(\Delta + 1)(n - 1)$.*

Proof. There are $\tau_{\mathcal{G}}$ snapshots, each one can be turned into a temporal graph that uses up to $n - 1$ nominal labels tilted in $\Delta + 1$ different ways. ☐

3.2.2 Saturation technique: "non-strict" → "strict"

As already explained, the dilation transformation described above makes it possible to transform any temporal graph in the non-strict setting into a proper graph, which is *de facto* included in both the strict and non-strict setting. Thus, it can be seen as a support-preserving transformation from the non-strict setting to the strict setting, at the cost of a significant blow-up of the lifetime. In this section, we present a weaker transformation called the *saturation* method. This transformation is only reachability-preserving, but it keeps the lifetime constant. An additional benefit is that it is pretty simple. A similar technique was used in [5] to test the temporal connectivity of a temporal graph with non-strict journeys.

Theorem 1. *Let $\mathcal{G}$ be a temporal graph with n vertices, m contacts and a lifetime of size τ, considered in the non-strict setting. There exists a temporal graph $\mathcal{H}$ with n vertices, lifetime τ and at most $(n(n+1)\tau)/2$ contacts in the strict setting that results in the same reachability graph.*

Proof. Let $\mathcal{G}$ be seen as a sequence of snapshots $G_1, ..., G_\tau$. The transformation consists of transforming independently every snapshot G_i of $\mathcal{G}$ by turning every path of G_i into an edge. In other words, turning each snapshot G_i into its own (path-based) transitive closure. The resulting graph $\mathcal{H} = H_1, ..., H_\tau$ has the same lifetime as $\mathcal{G}$ and the same set of vertices. We will now prove that there is a (non-strict) journey from u to v in $\mathcal{G}$ if and only if there is a strict journey from u to v in $\mathcal{H}$.

(→) Let j be a journey in $\mathcal{G}$. If any part of j uses consecutive edges at the same time step t (say, from a to b), then there is a corresponding path in the snapshot G_t, implying an edge ab in H_t, thus, this part of j can be replaced by a contact $(\{a, b\}, t)$ in $\mathcal{H}$. Repeating the argument implies a strict journey.

(←) Let j' be a journey in $\mathcal{H}$. By construction of $\mathcal{H}$, for any contact $(\{a, b\}, t)$ in j', either the same contact already exists in $\mathcal{G}$, or there exists a path between a and b in G_t. If non-strict journeys are allowed, this path can replace the contact. Repeating the argument implies a non-strict journey. ☐

3.2.3 Semaphore technique: "strict" → "simple & proper" (happy)

In this section, we describe a transformation called the *semaphore technique*, which transforms any graph in the strict setting into a happy graph, while preserving the reachability between the vertices of the input graph, thus this transformation implies an induced-reachability equivalence. Our transformation is inspired by a reduction due to Bhadra and Ferreira [7] that reduces the CLIQUE problem to the problem of finding maximum components in temporal graphs. However, their reduction takes as input a graph that is (morally) simple, and produces temporal graphs that are neither simple nor proper. Thus, our transformation differs significantly.

Theorem 2. *Let $\mathcal{G}$ be a temporal graph with n vertices and m contacts in the strict setting. There exists a happy graph $\mathcal{H}$ with $n+2m$ vertices and $4m$ edges and a mapping $\sigma : V_{\mathcal{G}} \to V_{\mathcal{H}}$ such that $(u,v) \in \mathcal{R}(\mathcal{G})$ if and only if $(\sigma(u),\sigma(v)) \in \mathcal{R}(\mathcal{H})$.*

Proof. Intuitively, the transformation consists of turning every contact of $\mathcal{G}$, say $x = (\{u,v\},t_x)$, into a "semaphore" gadget in $\mathcal{H}$ that consists of a copy of u and v, plus two auxiliary vertices u_x and v_x linked by 4 edges $\{u,u_x\}$, $\{u_x,v\}$, $\{u,v_x\}$, $\{v_x,v\}$, whose labels create a journey from u to v through u_x, and from v to u through v_x. The labels are chosen in such a way that these journeys can replace x for the composition of journeys in $\mathcal{H}$. For simplicity, our construction uses fractional label values, which can subsequently be renormalized into integers. A basic example is shown in Figure 4.

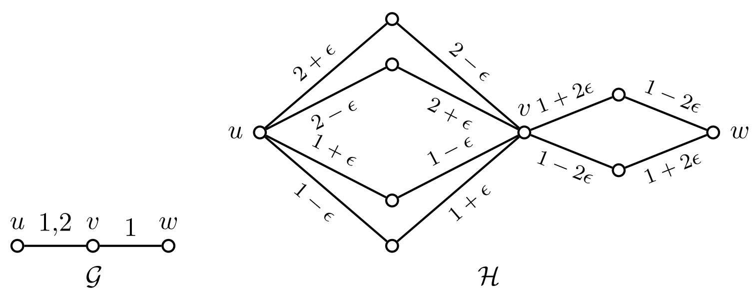


Figure 4: The semaphore technique, turning a non-proper graph $\mathcal{G}$ (in the strict setting), into a happy graph $\mathcal{H}$ whose reachability preserves the relation among original vertices.

Since we consider strict journeys in $\mathcal{G}$, if two adjacent time edges share the same time label, it should be forbidden to take the two of them consecutively. To ensure this, the time labels of $\{u,u_x\},\{u_x,v\},\{u,v_x\},\{v_x,v\}$ are respectively $t_x - \epsilon$, $t_x + \epsilon, t_x + \epsilon$, $t_x - \epsilon$ (with $0 < \epsilon < 1/2$), enabling the journeys from u to v and from v to u without making two such journeys composable if the two original labels are the same. The created edges have a single label, but the graph might not yet be proper. To ensure properness, we tilt slightly the time labels by multiples of ϵ, in a similar spirit as in the dilation technique presented above. More precisely, consider a proper edge-coloring of the footprint of $\mathcal{G}$ using $\Delta + 1$ colors in $\{1, ..., \Delta + 1\}$ (where Δ is the maximum degree in the footprint), such a coloring being guaranteed by Vizing's theorem. For each edge e of the footprint, note c_e its color. Now the time labels in $\mathcal{H}$ associated to $x = (e,t_x)$ are $t_x - c_e\epsilon$, $t_e + c_e\epsilon$ with $0 < \epsilon < \frac{1}{2(\Delta+1)}$.

The semaphore gadget is applied for each contact of $\mathcal{G}$ with the corresponding color, so if there was n vertices and m time edges in $\mathcal{G}$, then $\mathcal{H}$ will have $n + 2m$ vertices and $4m$

time edges. It is easy to see that $\mathcal{H}$ is now simple and proper. We will now prove that, for any u and v, u can reach v in $\mathcal{G}$ if and only if u can reach v in $\mathcal{H}$.

($\rightarrow$) First note that for any pair of adjacent vertices $u, v \in V_{\mathcal{G}}$, we can go from u to v in $\mathcal{H}$ by following one side of the "semaphore" and from v to u by following the other side. Furthermore, for any two $u, v \in V_{\mathcal{G}}$ such that there is a strict journey from u to v in $\mathcal{G}$, there is a sequence of edges $e_1, e_2, \ldots, e_k$ with labels $t_1, t_2, \ldots, t_k$ in $\mathcal{G}$ such that e_1 is incident to u, e_k is incident to v, e_i is adjacent to e_{i+1} and $t_i < t_{i+1}$. Then u can reach v in $\mathcal{H}$ by a sequence of edges with increasing labels $t_1 - c_{e_1}\epsilon, t_1 + c_{e_1}\epsilon, t_2 - c_{e_2}\epsilon, t_2 + c_{e_2}\epsilon, \ldots, t_k - c_{e_k}\epsilon, t_k + c_{e_k}\epsilon$.

($\leftarrow$) First note that from any original vertex a, seen in $\mathcal{H}$, we must first move to an auxiliary vertex b by the edge $e_1 = \{a, b\}$ and then to another original vertex c by edge $e_2 = \{b, c\}$. The labels on the edges must be $t - c\epsilon$ and $t + c\epsilon$ for some t, where t is the original time of the contact of $\mathcal{G}$ and c the color of the underlying edge. Then, we again reach an auxiliary vertex d from c at time $t' - c'\epsilon$, where $t + c\epsilon < t' - c'\epsilon$, and then another original vertex e at time $t' + c'\epsilon$. Since $t < t'$, we can go from a to e through c in $\mathcal{G}$ using contacts $(\{a, c\}, t)$ and $(\{c, e\}, t')$, respectively. Hence for all original u and v, if we have a temporal path from u to v in $\mathcal{H}$, then we can follow the above process multiple times to get a temporal path from u to v in $\mathcal{G}$. □

Observe that the semaphore technique is, in a quite relaxed way, also support-preserving, in the sense that a journey in $\mathcal{G}$ can be mapped into a journey in $\mathcal{H}$ that traverses the *original* vertices in the same order (and vice versa), albeit with auxiliary vertices in between.

3.3 Summary and discussions

Let S_1 and S_2 be two different settings, we define an order relation $\preceq$ so that $S_1 \preceq S_2$ means that for any graph $\mathcal{G}_1$ in S_1, one can find a graph $\mathcal{G}_2$ in S_2 such that $\mathcal{R}(\mathcal{G}_1) \simeq \mathcal{R}(\mathcal{G}_2)$. We write $S_1 \precnsim S_2$ if the containment is proper (i.e., there is a graph in S_2 whose reachability graph cannot be obtained from a graph in S_1). Finally, we write $S_1 \approx S_2$ if both sets of reachability graphs coincide. Several relations follow directly from containment among graph classes, e.g. the fact that simple graphs are a particular case of non-simple graphs. The above separations and transformations also imply a number of relations, and their combination as well. For example, proper graphs are contained both in the strict and non-strict settings, and since there is a transformation from non-strict graphs (in general) to proper graphs, we have the following striking relation:

Corollary 4. *"Proper" $\approx$ "non-strict".*

Similarly, combining the fact that "simple & non-strict" is strictly contained in "non-strict" (by Corollary 2), and there exists a reachability-preserving (in fact, support-preserving) transformation from "non-strict" to "proper", we also have that

Corollary 5. *"Simple & non-strict" $\precnsim$ "proper".*

Finally, the fact that there is a reachability-preserving transformation from "non-strict" to "strict" (the saturation technique), and some reachability graphs from "simple & strict" are unrealizable in "non-strict" (by Lemma 3), we also have

Corollary 6. *"Non-strict" $\precnsim$ "strict".*

A summary of the relations is shown in Figure 5, where green thick edges represent the transformations that are support-preserving, green thin edges represent transformations

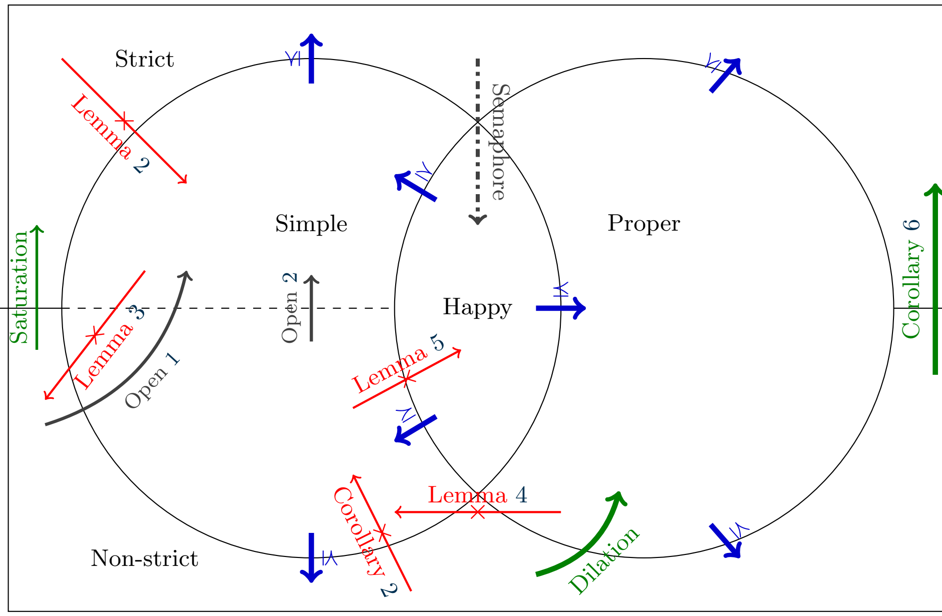


Figure 5: Separations, transformations, and inclusions among settings.

that are reachability-preserving, red edges with a cross represent separations (i.e. the impossibility of such a transformation), black dashed edges represent the induced-reachability-preserving transformation to happy graphs. Finally, inclusions of settings resulting from containment of graph classes are depicted by short blue edges. Some questions remain open. In particular,

Open question 1. *Does "non-strict" $\preceq$ "simple & strict"? In other words, is there a reachability-preserving transformation from the former to the latter?*

By Lemma 3, we know that both settings are not equivalent, but are they comparable? If not, a similar question holds for "simple & non-strict":

Open question 2. *Does "simple & non-strict" $\preceq$ "simple & strict"? In other words, is there a reachability-preserving transformation from the former to the latter?*

To conclude this section, Figure 6 depicts a hierarchy of the settings ordered by the above relation $\preceq$; i.e. by the sets of reachability graph they can achieve.

4 Strengthening existing results to happy graphs

From the previous section, happy graphs are the least expressive setting. In this section, however, we argue that they remain expressive enough to strengthen existing negative results for at least two well-studied problems. First, we show that the construction from [4] can be made happy, which implies that $o(n^2)$-sparse spanners do not always exist in happy graphs. We also show that the reduction from clique to temporal component from [7] can be made happy, which implies that temporal component is NP-complete even in happy graphs (for both open and closed components). Finally, to further motivate studies on happy graphs, we list a few open questions.

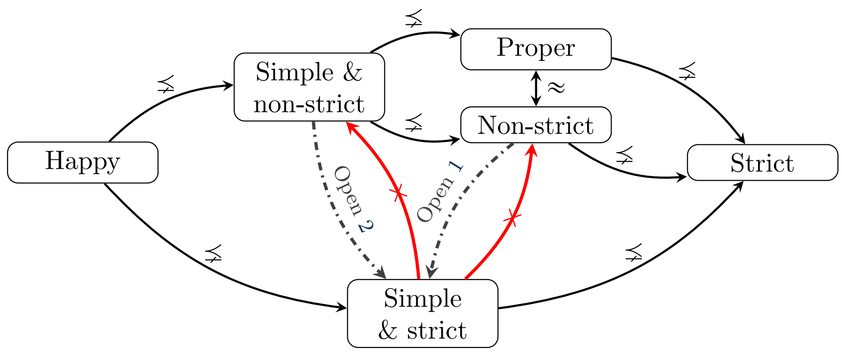


Figure 6: Ordering of temporal graph settings in terms of reachability graphs.

4.1 Temporal Component remains hard

The temporal component problem can be defined as follows in temporal graphs.

> Temporal Component
> Input: A temporal graph $\mathcal{G} = (V, E, \lambda)$, an integer k
> Output: Is there a set $V' \subseteq V$ of size k in $\mathcal{G}$ such that all vertices of V' can reach each other by a temporal path?

Due to the non-transitive nature of reachability, two versions are typically considered, depending on whether the vertices of V' can rely (open version) or not (closed version) on vertices outside of V' for reaching each other.

In [7], Bhadra and Ferreira show that both versions of the problem are NP-complete. Interestingly, although the authors consider a strict setting in the paper, their construction works indistinctly for both the strict or the non-strict setting. However, it is neither proper nor simple. In this section, we explain how to adapt their construction to happy graphs. Our reduction works for both the open and closed version, for the simple fact that it produces an instance where the maximum open and closed components are the same.

Let (G, k) be an instance of the clique problem, where G is a static undirected graph and k some integer, the reduction in [7] transforms G into a temporal graph $\mathcal{G}$ as follows. The first step of the transformation corresponds to a simplified version of the semaphore technique, where two auxiliary vertices are created for each pair of neighbors u and v in G, such that u can reach v in $\mathcal{G}$ through one of these vertices and v can reach u through the other, using labels 2 on the first edge and 3 on the second (on both sides). Observe that two adjacent semaphores have labels which are not proper. The second step is to connect all pairs of auxiliary vertices x and y using an edge xy with two labels 1 and 4 for each pair (thus $\mathcal{G}$ is not simple). The purpose of these contacts is to make all auxiliary vertices reachable from each other, and to create journeys between each auxiliary vertex and each original vertex (both ways). As a result, an SCC of size $2m + k$ exists in $\mathcal{G}$ (where m is the number of edges of G) if and only if a clique of size k exists in G.

Theorem 3. Temporal Component *is NP-complete in happy graphs.*

Proof. Our proof consists of making the transformation from [7] both simple and proper, while preserving the size of the maximum component within (which is the same for the open and closed versions). First, observe that the non-proper labels of the semaphores in $\mathcal{G}$ can easily be turned into proper labels by *tilting* the labels in the same way as explained in the *semaphore technique* (Theorem 2 on page 13), namely, by coloring properly the edges of the footprint of $\mathcal{G}$, and adding the corresponding multiple of ϵ to every label. Since the quantity added to each label is less than 1, the reachability among original vertices, and between original and auxiliary vertices, is unaffected. The conversion of the labels 1 and 4 between auxiliary vertices is slightly more complicated. These vertices in $\mathcal{G}$ form a clique, each edge of which have labels 1 and 4. First, using Lemma B.1 in [4], any clique of $2m$ vertices may be decomposed into m hamiltonian paths. We need only 4 such paths for the construction, which is guaranted as soon as $m \geqslant 4$ (our adaptation does not need to hold for smaller graphs in order to conclude that the problem is NP-complete). Thus, take four edge-disjoint hamiltonian paths among auxiliary vertices. We will use two of them, say p_1 and p_2 to replace the contacts having label 1 (the same technique applies for label 4). Pick a vertex u in p_1 and assign time labels to p_1 so that all vertices in p_1 can reach u through ascending labels (towards u). Then proceed similarly in p_2 with ascending labels *from* u towards all the vertices of p_2. Choose these labels so that they remain sufficiently close to 1 and do not interfere with the rest of the construction. Proceed similarly for the two other hamiltonian paths with respect to label 4. The resulting construction is happy and preserves the journeys between auxiliary vertices, while preserving the composability of journeys with the rest of the construction. □

4.2 Happy graphs do not always admit $o(n^2)$-sparse spanners

In [4] (Theorem 3.1), Axiotis and Fotakis construct an infinite family of temporal graphs in the "simple & non-strict" setting that does not admit a $o(n^2)$-sparse spanner. The goal of this section is to show that their construction can be strengthened to happy graphs.

The construction $\mathcal{G}$ in [4] consists of three parts of n vertices each (thus $N = 3n$ vertices in total), namely a clique of vertices $A = a_1, ..., a_n$; an independent set $H = h_1, ..., h_n$; and a set $M = m_1, ..., m_n$ of additional vertices. The idea is to make every edge of A critical to provide connectivity among some vertices of H, so that removing any of these edge breaks temporal connectivity and every spanner thus contains $\Theta(n^2)$ edges. The purpose of the vertices in M is only to make the rest of $\mathcal{G}$ temporally connected without affecting these relations between A and H. In this construction, every edge receives a single label, so $\mathcal{G}$ is already simple. However, the inner labeling of the clique A is not proper. We claim that this labeling can be made proper without affecting the main properties of the construction. The following statement is identical to Theorem 3.1 in [4], except that the adjective happy is inserted.

Theorem 4. *For any even $n \geqslant 2$, there is a happy connected temporal graph with $N = 3n$ vertices, $\frac{n(n+9)}{2} - 3$ edges and lifetime at most $\frac{n(n+5)}{2} - 1$, so that the removal of any subset of $5n$ edges results in a disconnected temporal graph.*

Proof (sketch). The complete construction from [4] is not presented here in detail. However, the fact that it can be made proper relies on a simple observation. In [4], the clique A is decomposed into $\frac{n}{2}$ hamiltonian paths $p_1, p_2, ..., p_{n/2}$, each of which is assigned label i. For every path p_i, vertices h_{2i-1} and h_{2i} are connected to the endpoints of p_i (one on each side), and the main requirement is that this path is the *only* way for h_{2i-1} to reach h_{2i}. Interestingly, although every path p_i is non-strict in [4] and thus could be travelled in both directions, it turns out that only one direction is needed, because h_{2i} can reach h_{2i-1} (for every i) using temporal paths outside of the clique. Therefore, our adaptation consists of assigning to every path i a strictly increasing sequence of labels, while shifting all the larger labels of the graph appropriately, so that all other temporal paths are unaffected and $\mathcal{G}$ becomes proper. The full proof would require a complete description of the construction in [4], which would be identical up to the above change. □

4.3 Further questions

To conclude this section, we state a few open questions related to spanners in happy graphs. The first question is structural, namely,

Open question 3. *Do happy cliques always admit $O(n)$-sparse spanners? If so, do they admit spanners of size $2n-3$?*

It was shown in [17] that happy cliques (or alternatively, all temporal cliques in the non-strict setting) always admit spanners of size $O(n \log n)$, but no counterexamples were found so far that rule out size $O(n)$.

On the algorithmic side, two independent results establish that Min-Label Spanner is hard in temporal graphs [4, 2]. However, the proofs in these papers rely on constructions which are not happy, and it is not clear that these constructions can be strengthened to happy graphs in a similar way as the above problems. Thus,

Open question 4. *Is* Min Spanner *tractable in happy graphs?*

As already explained, both the Min-Edge Spanner and Min-Label Spanner versions of the problem coincide in simple graphs (and thus in happy graphs), which makes them a single problem.

5 Concluding remarks

In this paper, we explored the impact of three particular aspects of temporal graphs: *strictness*, *properness*, and *simpleness*. Comparing their expressivity in terms of reachability graphs, we showed that these aspects really matter and that separations exist between the expressivity of some settings, while others can be shown equivalent through transformations. Then, we focused on the simplest model (happy graphs), where all these distinctions vanish, and showed that this model still captures interesting features of general temporal graphs. Our results imply a few striking facts, such as the fact that the "proper" setting is as expressive as the "non-strict" setting. Some relations remain unknown, in particular, it is open whether the "non-strict" setting is comparable to "simple & strict" setting. Finally, despite their extreme simplicity, several basic questions remain open on happy graphs, which we think makes them a natural target for further studies. We conclude by stating a few questions of more general scope, related to the present paper.

Open question 5 (Realizability of a reachability graph)**.** *Given a static digraph, how hard is it to decide whether it can be realized as the reachability graph of a temporal graph?*

The structural analog of this question could be formulated as follows

Open question 6 (Characterization of the reachability graphs)**.** *Characterize the set of static directed graphs that are the reachability graphs of some temporal graph.*

Questions 5 and 6 can be declined into several versions, one for each setting. Finally, the work in this paper focused on *undirected* temporal graphs. It would be interesting to see if the expressivity of *directed* temporal graphs shows similar separations and transformations.

Open question 7 (Directed temporal graphs)**.** *Does the expressivity of directed temporal graphs admit similar separations and transformations as in the undirected case?*

References

[1] Eleni C Akrida, Jurek Czyzowicz, Leszek Gasieniec, Łukasz Kuszner, and Paul G Spirakis. Temporal flows in temporal networks. *Journal of Computer and System Sciences*, 103:46–60, 2019.

[2] Eleni C. Akrida, Leszek Gasieniec, George B. Mertzios, and Paul G. Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61(3):907–944, 2017.

[3] Karine Altisen, Stéphane Devismes, Anaïs Durand, Colette Johnen, and Franck Petit. On implementing stabilizing leader election with weak assumptions on network dynamics. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 21–31, 2021.

[4] Kyriakos Axiotis and Dimitris Fotakis. On the size and the approximability of minimum temporally connected subgraphs. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 149:1–149:14, 2016.

[5] Matthieu Barjon, Arnaud Casteigts, Serge Chaumette, Colette Johnen, and Yessin M Neggaz. Testing temporal connectivity in sparse dynamic graphs. *arXiv preprint arXiv:1404.7634*, 2014.

[6] Hervé Baumann, Pierluigi Crescenzi, and Pierre Fraigniaud. Parsimonious flooding in dynamic graphs. *Distributed Computing*, 24(1):31–44, 2011.

[7] Sandeep Bhadra and Afonso Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 259–270. Springer, 2003.

[8] Davide Bilò, Gianlorenzo D'Angelo, Luciano Gualà, Stefano Leucci, and Mirko Rossi. Sparse temporal spanners with low stretch. *arXiv preprint arXiv:2206.11113*, 2022.

[9] Quentin Bramas and Sébastien Tixeuil. The complexity of data aggregation in static and dynamic wireless sensor networks. In *Symposium on Self-Stabilizing Systems*, pages 36–50. Springer, 2015.

[10] Luiz FA Brito, Marcelo K Albertini, Arnaud Casteigts, and Bruno AN Travençolo. A dynamic data structure for temporal reachability with unsorted contact insertions. *Social Network Analysis and Mining*, 12(1):1–12, 2022.

[11] Filippo Brunelli, Pierluigi Crescenzi, and Laurent Viennot. On computing pareto optimal paths in weighted time-dependent networks. *Inf. Process. Lett.*, 168:106086, 2021.

[12] B. Bui-Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02):267–285, 2003.

[13] Richard T. Bumby. A problem with telephones. *SIAM Journal on Algebraic Discrete Methods*, 2(1), 1979.

[14] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

[15] Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83(9):2754–2802, 2021.

[16] Arnaud Casteigts, Ralf Klasing, Yessin M Neggaz, and Joseph G Peters. Computing parameters of sequence-based dynamic graphs. *Theory of Computing Systems*, 63:394–417, 2019.

[17] Arnaud Casteigts, Joseph G. Peters, and Jason Schoeters. Temporal cliques admit sparse spanners. *Journal of Computer and System Sciences*, 121:1–17, 2021.

[18] Arnaud Casteigts, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp thresholds in random simple temporal graphs. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–326. IEEE, 2022.

[19] Václav Chvátal and J Komlós. Some combinatorial theorems on monotonicity. *Canadian Mathematical Bulletin*, 14(2):151–157, 1971.

[20] Alessio Conte, Pierluigi Crescenzi, Andrea Marino, and Giulia Punzi. Enumeration of sd separators in dags with application to reliability analysis in temporal graphs. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[21] G Di Luna, Stefan Dobrev, Paola Flocchini, and Nicola Santoro. Distributed exploration of dynamic rings. *Distributed Computing*, 33(1):41–67, 2020.

[22] Jessica Enright, Kitty Meeks, George B Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[23] Jessica Enright, Kitty Meeks, and Hendrik Molter. Counting temporal paths. *arXiv preprint arXiv:2202.12055*, 2022.

[24] Thomas Erlebach and Jakob T Spooner. Parameterized temporal exploration problems. In *1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[25] 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.

[26] Carlos Gómez-Calzado, Arnaud Casteigts, Alberto Lafuente, and Mikel Larrea. A connectivity model for agreement in dynamic systems. In *European Conference on Parallel Processing*, pages 333–345. Springer, 2015.

[27] 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.

[28] 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.

[29] Jayadev Misra and David Gries. A constructive proof of vizing's theorem. In *Information Processing Letters*. Citeseer, 1992.

[30] Ariel Orda and Raphael Rom. Minimum weight paths in time-dependent networks. *Networks*, 21(3):295–319, 1991.

[31] Léo Rannou, Clémence Magnien, and Matthieu Latapy. Strongly connected components in stream graphs: Computation and experimentations. In *International Conference on Complex Networks and Their Applications*, pages 568–580. Springer, 2020.

[32] 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.

[33] John Whitbeck, Marcelo Dias de Amorim, Vania Conan, and Jean-Loup Guillaume. Temporal reachability graphs. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 377–388, 2012.