On Inefficiently Connecting Temporal Networks

Esteban Christiann
École normale supérieure Paris Saclay, 91190 Gif-sur-Yvette, France

Eric Sanlaville
Université Le Havre Normandie, Univ Rouen Normandie, INSA Rouen Normandie, Normandie Univ, LITIS UR 4108, F-76600 Le Havre, France

Jason Schoeters
University of Cambridge, United Kingdom

Abstract

A temporal graph can be represented by a graph with an edge labelling, such that an edge is present in the network if and only if the edge is assigned the corresponding time label. A journey is a labelled path in a temporal graph such that labels on successive edges of the path are increasing, and if all vertices admit journeys to all other vertices, the temporal graph is temporally connected. A temporal spanner is a sublabelling of the temporal graph such that temporal connectivity is maintained. The study of temporal spanners has raised interest since the early 2000's. Essentially two types of studies have been conducted: the positive side where families of temporal graphs are shown to (deterministically or stochastically) admit sparse temporal spanners, and the negative side where constructions of temporal graphs with no sparse spanners are of importance. Often such studies considered temporal graphs with happy or simple labellings, which associate exactly one label per edge. In this paper, we focus on the negative side and consider proper labellings, where multiple labels per edge are allowed. More precisely, we aim to construct dense temporally connected graphs such that all labels are necessary for temporal connectivity. Our contributions are multiple: we present exact or asymptotically tight results for basic graph families, which are then extended to larger graph families; an extension of an efficient temporal graph labelling generator; and overall denser labellings than previous work, whether it be global or local density.

2012 ACM Subject Classification Networks

Keywords and phrases Network design principles, Network dynamics, Paths and connectivity problems, Branch-and-bound

Digital Object Identifier 10.4230/LIPIcs.SAND.2024.8

Related Version *Full Version*: `https://arxiv.org/abs/2312.07117`

Supplementary Material *Software (Source Code)*: `https://gitlab.com/echrstnn/PTGen`
archived at `swh:1:dir:a40ecefd5b1d1554ed3ee9c6aed8fad4bbdf43a9`

Funding *Eric Sanlaville*: project ANR-22-CE48-0001 (TEMPOGRAL)
Jason Schoeters: DyNet RIN Tremplin Région Normandie 2020-2022 & Leverhulme Trust International Professorship in Neuroeconomics

Acknowledgements We thank the referees for their careful reading and constructive comments which significantly improved the presentation of these results.

1 Introduction

A temporal graph is a graph which can evolve over time, through the appearing and/or disappearing of edges. Numerous classical graph problems and parameters have been extended to temporal graphs, such as colouring, connected components, maximum matchings, and independent sets [19, 28, 30, 33]. In temporal graphs, connectivity may become very poor when considering the graph at every distinct time step, but the graph may still be connected when considering connectivity over time. Indeed, temporal connectivity is motivated through many contexts in which temporal graphs naturally arise, most notably the context of swarms of mobile entities with distance-based communication capabilities (drone networks, insect colonies, and, particularly useful during the COVID-19 pandemic: people) [12, 14, 15, 21]. This temporal connectivity has since redefined classical connectivity problems, such as (temporal) dominating sets, and (temporally) connected components, and particularly interesting concerning this paper: (temporal) spanners [4, 6, 10, 24].

After presenting a wide range of interesting changes and results concerning typical graph problems with temporal paths instead of paths, Kempe, Kleinberg, and Kumar discuss further interesting questions [20]. One of these is whether a temporally connected graph can always be sparsified (that is, if labels can be removed) so as to obtain a "sparse" remaining structure maintaining temporal connectivity. Such a structure is later called a temporal spanner. Note that the static graph analogue would be asking whether a connected graph always admits a spanning tree, which is of course always the case. They follow up with a preliminary negative result, stating that some temporal graphs do not admit a linear size spanner (hypercube graphs with each edge labelled with the corresponding dimension). The real question then became whether dense temporal graphs could always admit a sparse spanner, the intuition being that there exists many more ways to potentially sparsify a dense graph. The question remained open for many years, until Axiotis and Fotakis answered in the negative: they construct a non-trivial dense temporal graph in which some labels may be removed but prove that a dense part has to remain to ensure temporal connectivity. [2] A couple of years afterwards, a complementing positive result was presented by Casteigts, Peters, and Schoeters: any temporal complete graph always admits a sparse spanner [10]. Following these, more papers surfaced related to temporal spanners: sharp thresholds on the density of random temporal graphs to asymptotically almost surely admit particular sparse spanners; positive and negative results regarding spanners which have a limited stretch, as well as on temporal spanners which are blackout-resistant [5, 6, 11].

Another topic of interest in temporal graph theory is that of temporal network design, where instead of analysing a given temporal graph, one would like to design a temporal graph with some desired property or decide such a temporal graph does not exist. In most works on temporal network design, the graph itself is given and a corresponding labelling needs to constructed. One of the earliest such design problems was to create a gossip protocol, that is, a schedule of pairwise communications between n agents, each having some piece of information which can be transferred over successive communications, such that at the end of the schedule, all agents are up to date with all the information. It is natural to minimise the number of communications (*e.g.* the total cost of phone calls), and thus some tight results arise with protocols using $2n-3$ communications, with the idea being to gather all information to some agent and then broadcast the information out again, designing essentially a temporal in-tree and a temporal out-tree *resp.* For more information, see survey [18]. More recently in [29], Mertzios *et al.* reconsider and extend this work as a temporal graph design problem. Direct results from gossiping apply, but more importantly, they include other restrictions on the labelling, such as a maximum lifetime *i.e.* the labels cannot be greater than some value, which was further investigated in [22]. Also, two measures of density, both of interest for this paper, are defined regarding a temporal graph: the *temporal cost*, being the total amount of labels; and the *temporality*, being the maximum amount of labels on some edge. The former is a global density measure, and the latter a local one.

In this paper, we combine the study of temporal spanners and of temporal graph design, by designing dense temporal graphs such that each label is necessary for temporal connectivity. As opposed to most previous work we will not restrict ourselves to happy or simple labellings

Measure \ Graph class	Any class	`Trees` (`Connected`)	`Cycles` (`Hamiltonian`)	`Cacti` (`Circumference` c)
Maximum temporal cost T^+	$\leq n^2 - n - 1$ Theorem 2	$2n - 3$ Theorem 5	$\geq \frac{1}{4}n^2 + \frac{3}{4}$ Theorem 7	$\geq \frac{1}{4}c^2 + 2(n-c)$ omitted (see [13])
Maximum temporality τ^+	$\leq n - 1$ Theorem 2	2 Theorem 4	$\geq \lceil \frac{1}{2}n \rceil$ Theorem 8	$\geq \lceil \frac{1}{2}c \rceil$ omitted (see [13])

Figure 1 Main results of our density measures on specific classes of graphs: $\leq$ indicates an upper bound, $\geq$ indicates a lower bound. Results for `Trees`, and by extension `Connected` graphs, are tight.

(one label per edge) but instead extend to consider proper labellings (multiple labels per edge allowed). This is a double-edged sword: on the one hand this intuitively may allow for much denser labellings, but on the other hand, a combinatorial explosion on the amount of possible labellings occurs implying algorithmics may be more difficult in this setting. In a sense, we are interested in designing the most inefficient temporal networks possible. Outside of the already established applications of temporal spanners in related work, the negative results in particular can have direct implications concerning adversarial behaviour in temporal network game theory and the potential waste of temporal and structural resources [27,32]. Lastly, a slowly temporally connected network may allow for time to detect any anomalies/viruses before the whole network is infected, while not hindering the supposedly essential connectivity of the network, and may have applications for fraud detection in financial transactions [31].

In short, throughout the paper, we will steadily answer both of the following questions:

- What is the densest temporal graph overall?
- Given a graph class, what is the densest labelling all graphs can attain?

1.1 Contributions

First, in Section 2, we give standard graph theory and temporal graph theory notation and define our setting as well as a global and a local density measure: maximum temporal cost T^+ and maximum temporality τ^+ respectively. Lower bounds from the literature and upper bounds through analysis are presented. Then, in Section 3, we focus on tree graphs for which we obtain tight results through an argument on bridge edges. These results do not beat aforementioned lower bounds however. In Section 4, we focus on cycles, partly due to a lower bound being a labelling on cycles, which shows promise for obtaining even denser labellings. For this, we decide to extend labelling generator STGen from [7] so as to fit to our setting and specifically to cycles. After executing it on small cycles, we obtain the intuition for a complex labelling which beats the lower bounds for local density by a factor of 1.5, and for global density by exactly 1 label. A non-trivial proof is given to show that all labels of this labelling are indeed necessary and that the resulting temporal graph is temporally connected, using a representation of temporal graphs called link streams and by reasoning on journeys which are necessary. A summary of our results is presented in Figure 1, where n is the number of vertices of a graph. We discuss and extend results in Section 5, and conclude.

Due to the page limit, technical proofs (marked with a $\star$) were moved to the appendix and other parts were omitted. For these, the reader is referred to the full version on arXiv [13].

2 Preliminaries

In this paper, all graphs are simple and undirected (except for the reachability graph defined below). A temporal graph is a tuple (G, λ) with graph $G = (V, E)$, called the footprint or underlying graph, and edge labelling $\lambda : E \to 2^{\mathbb{N}}$. The labels correspond to when the edges are present over the lifetime of the temporal graph. A pair (e, ℓ) with $e \in E$ and $\ell \in \lambda(e)$ is called a time edge. Reachability in temporal graphs is defined through temporal paths, also called journeys, which are adjacent time edges $j = (c_1, c_2, ..., c_k)$ such that for all $c_i = (e_i, \ell_i)$ with $i \in [2, k]$ we have that $\ell_i > \ell_{i-1}$. We say u can reach v, or v can be reached by u, if there exists a journey from u to v. A journey $\mathcal{J}$ is said to cover a set of vertices V' if for all vertices v in V', v is part of some time edge of $\mathcal{J}$. A temporal graph $\mathcal{G}' = (G', \lambda')$ is a temporal subgraph of $\mathcal{G} = (G, \lambda)$ if G' is a subgraph of G and λ' a sublabelling of λ. For a label ℓ of temporal graph $\mathcal{G}$, $\mathcal{G}^{-\ell}$ corresponds to the temporal subgraph of $\mathcal{G}$ without label ℓ (if other labels exist in $\mathcal{G}$ with the same value, then these remain).

A temporal branching $\mathcal{B} = (T, \lambda)$ with root r is a tree T with $|\lambda| = n - 1$ such that vertex r can reach all vertices. A temporal branching $\mathcal{B} = (T, \lambda')$ with root v of a temporal graph $\mathcal{G} = (G, \lambda)$ is a temporal subgraph of $\mathcal{G}$ which is a temporal branching, and it is spanning if $V(T) = V(G)$. The reachability graph $R(\mathcal{G})$ is defined on the same vertex set as $\mathcal{G}$ and an arc exists from u to v if and only if u can reach v in $\mathcal{G}$. If all vertices can reach all other vertices in $\mathcal{G}$, we say $\mathcal{G}$ is temporally connected. Note that a temporal graph is temporally connected if and only if the corresponding reachability graph is complete (with arcs in both directions). From [8], two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are reachability-equivalent[1] if reachability graphs $R(\mathcal{G}_1)$ and $R(\mathcal{G}_2)$ are isomorphic, denoted $\mathcal{G}_1 \overset{\mathrm{R}}{\simeq} \mathcal{G}_2$.

A labelling is proper when no incident edges share a same label. In the rest of the paper, all labellings are supposed proper, unless specifically stated otherwise. Using terms from [1], a label ℓ in a temporal graph $\mathcal{G}$ is redundant if and only if it can be removed from $\mathcal{G}$ without reducing reachability, *i.e.* $\mathcal{G} \overset{\mathrm{R}}{\simeq} \mathcal{G}^{-\ell}$. Conversely, a label ℓ of $\mathcal{G}$ is necessary if and only if $\mathcal{G} \overset{\mathrm{R}}{\not\simeq} \mathcal{G}^{-\ell}$. If a labelling contains only necessary labels, we call it a minimal labelling. We call a temporal graph with a proper (*resp.* minimal) labelling a proper (*resp.* minimal) temporal graph.

In [29], the authors defined two measures of density for a temporal graph $\mathcal{G}$: the temporal cost $T(\mathcal{G})$, which is the total amount of labels in $\mathcal{G}$; and the temporality $\tau(\mathcal{G})$ which is the maximum amount of labels on an edge, among all edges. The former is intended as a global density measure, whereas the latter is more of a local one, potentially of interest for example in distributed or parallel computing. We adapt temporal cost in the following manner. The three types of maximum temporality are defined analogously.

- Let $T^+(G)$ be the maximum temporal cost of graph G, *i.e.* the maximum temporal cost $T(\mathcal{G} = (G, \lambda))$ of all proper minimal labellings λ such that $\mathcal{G}$ is temporally connected;
- Let $T^+(\texttt{Class})$ be the maximum temporal cost of graph class `Class`, *i.e.* the maximum value x such that for all graphs G of `Class`, $T^+(G) \geq x$;
- Let T^+ be the maximum temporal cost, *i.e.* the maximum temporal cost $T^+(G)$ among all graphs G on n vertices.

We study three simple graph classes in this work, being `Trees` (Section 3), `Cycles` (Section 4), and `Cacti` (omitted, see [13]), and in Section 5 superclasses are discussed.

2.1 Upper and lower bounds on T^+ and τ^+

The upper bounds are both obtained through the following fact:

▶ **Lemma 1.** *A minimal temporally connected graph $\mathcal{G}$ equals the union of any n spanning temporal branchings with distinct roots of $\mathcal{G}$.*

[1] Originally *closure-equivalent*, but changed to reachability-equivalent for journal version (private message).

Proof. By contradiction, suppose that the union of some n spanning temporal branchings with distinct roots of $\mathcal{G}$ does not equal $\mathcal{G}$. This implies at least some label of $\mathcal{G}$ isn't part of the branchings, which means it can be removed without reducing reachability. However, $\mathcal{G}$ is minimal so no redundant labels exist which is a contradiction. ◀

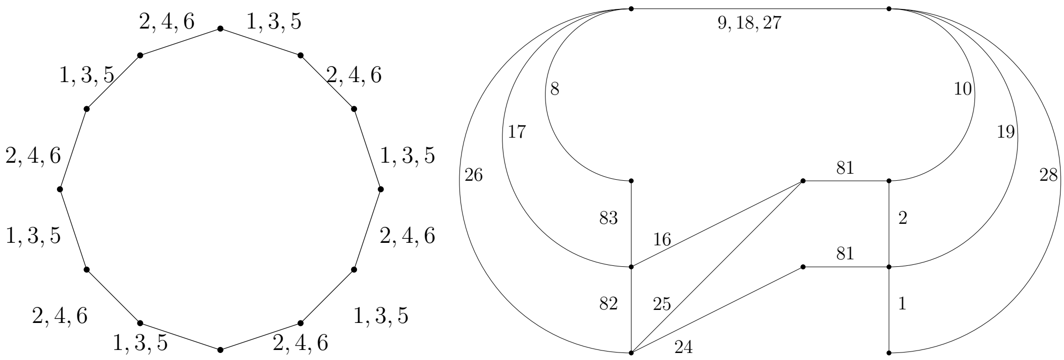
Lemma 1 allows us to reason on minimal temporally connected graphs through the corresponding spanning temporal branchings: for the largest possible maximum temporal cost T^+, consider the branchings to all be using distinct labels; whereas the branchings all using some same edge with distinct labels would result in the largest maximum temporality τ^+.

▶ **Theorem 2** ($\star$)**.** *The maximum temporal cost* $T^+ \leq n^2 - n - 1$ *and the maximum temporality* $\tau^+ \leq n - 1$.

Observe that the idea of considering n root-distinct temporal branchings is used in Observation 3 in [22] as well. We believe that their result could be improved slightly from $n(n-1)$ to $n(n-1)-1$ in a similar fashion as we do for the maximum temporal cost T^+.

Also in [22], Klobas et al. construct some minimal labellings which are strict, meaning journeys are allowed to traverse at most one edge per time step. Among these labellings, the one given in Lemma 4 happens to be proper, giving lower bounds $T^+ \geq \frac{1}{4}n^2$ and $\tau^+ \geq \frac{1}{4}n$. The idea of the labelling is to label every other edge of an even cycle graph with all even labels up to $n/2$, and all other edges with all odd labels up to $n/2$. Let us refer to this labelling as the parity labelling (see Figure 2).

During the open question session of a Dagstuhl seminar on temporal graphs (see [9] for the report), some preliminary results of this work were presented, including a labelling of an ad-hoc graph giving lower bounds $T^+ \geq \frac{1}{18}n^2 + O(n)$ and $\tau^+ \geq \lfloor \frac{1}{3}n \rfloor$. The idea is to force journeys to go through the top edge using distinct labels. Let us refer to this labelling as the ad-hoc labelling (see Figure 2).



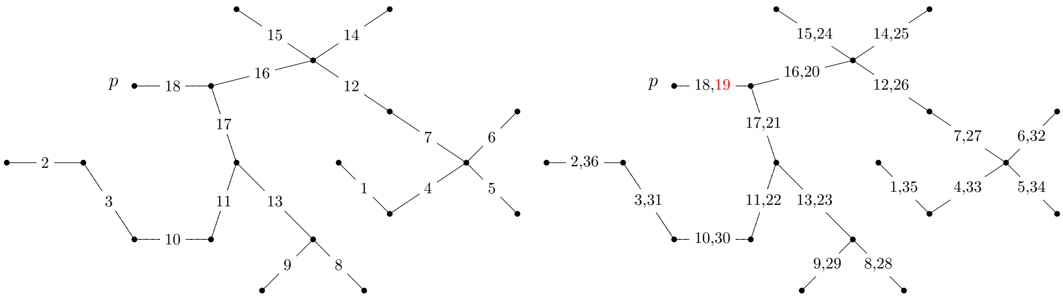
(a) Parity labelling from [22] for $n = 12$. **(b)** Ad-hoc labelling from [9] for $n = 10$.

Figure 2 Minimal labellings from the literature giving lower bounds on maximum temporal cost T^+ and maximum temporality τ^+.

Note that the parity labelling is denser regarding the temporal cost, but the ad-hoc labelling is denser regarding temporality. Some natural questions arise from these bounds. It seems that very sparse graphs (such as `Cycles`) can admit dense labellings, does that mean that other sparse graphs, such as `Trees`, can admit dense labellings as well? Can one do better than these labellings, and more specifically better in `Cycles`? These questions are answered in the following sections.

3 Tree graphs

In this section we prove the following labelling is densest possible for tree graphs, considering maximum temporal cost T^+ and maximum temporality τ^+. The labelling originates from gossiping strategies from e.g. [3, 17] and has been used in temporal graph theory papers such as in Theorem 2 in [1] and the pivot technique in [10]. We refer to it as the pivot labelling, and define it as follows (see also Figure 3). Select an arbitrary pivot vertex p and construct journeys from all other vertices towards p, using the reverse breadth-first search order. Then, using the breadth-first search order, add journeys from p to all other vertices. The earliest label of the second BFS is removed.



(a) First (reverse) breadth-first search labelling. **(b)** Adding the second breadth-first search labelling.

Figure 3 The pivot labelling of an example tree graph. A first labelling converging to pivot vertex p is shown, which is then complemented by a second broadcasting labelling from p. Label 19 (shown in red) is redundant and removed.

The resulting temporal graph is temporally connected since by design all vertices can reach pivot vertex p at time $n-1$, and starting at time $n-1$, vertex p can reach all vertices. It is also a minimal labelling since removing any label $\leq n-1$ on a path from a leaf vertex f to p reduces the reachability of f, and removing any label $> n-1$ makes it so f cannot be reached by some other leaf vertex.

As stated by Theorem 2(a) in [1], the pivot labelling thus trivially gives the lower bound $T^+(\texttt{Trees}) \geq 2n-3$. Also, it trivially gives the lower bound of $\tau^+(\texttt{Trees}) \geq 2$. To prove these are tight, *i.e.* no denser labellings exist, we first present a lemma focusing on bridge edges (edges which disconnect the graph if removed), and then apply it on tree graphs.

▶ **Lemma 3.** *For any bridge edge e of graph G and any minimal labelling λ such that $\mathcal{G} = (G, \lambda)$ is temporally connected, λ can assign at most two labels to e.*

Proof. Consider a temporally connected graph $\mathcal{G}$ with bridge edge $e = \{u, v\}$, separating $\mathcal{G}$ into two temporal subgraphs $\mathcal{G}_1$ (with vertex u) and $\mathcal{G}_2$ (with v). Suppose by contradiction that the labelling λ of $\mathcal{G}$ assigns more than two labels to edge e, say $k > 2$ labels $\ell_1, \ell_2, ..., \ell_k$, and that this labelling is minimal. Define t_u^- to be the earliest time at which all vertices in $\mathcal{G}_1$ are able to reach u. Similarly, define t_v^+ to be the latest time at which all vertices in $\mathcal{G}_2$ can be reached by v. Since $\mathcal{G}$ is temporally connected, there exists some label ℓ_i of e such that $t_u^- < \ell_i < t_v^+$. Keeping ℓ_i is thus sufficient for maintaining reachability from all vertices in $\mathcal{G}_1$ to all vertices in $\mathcal{G}_2$. A symmetrical argument can be used to find label ℓ_j which is sufficient for maintaining reachability from all vertices in $\mathcal{G}_2$ to all vertices in $\mathcal{G}_1$. Together, ℓ_i and ℓ_j are thus sufficient for reachability concerning journeys using edge e, and edge e can trivially be ignored for reachability between vertices in $\mathcal{G}_1$ (*resp.* $\mathcal{G}_2$). This results in all other labels on edge e being redundant, which is a contradiction since it is minimal. ◀

▶ **Theorem 4.** $\tau^+(\mathtt{Trees}) = 2$.

Proof. Tree graphs contain only bridge edges, so by Lemma 3 no edge of a tree graph can have more than two labels in a minimal labelling, and this is attained by the pivot labelling. ◀

Note that Theorem 4 implies that $T^+(\mathtt{Trees}) \leq 2n - 2$, which is only off by 1 from the lower bound. We finish this section with proving the latter to be tight.

▶ **Theorem 5** ($\star$). $T^+(\mathtt{Trees}) = 2n - 3$.

As a side note, we remark that the maximum temporal cost (*resp.* temporality) of tree graphs corresponds to the minimum temporal cost (*resp.* temporality) of tree graphs. In other words, all minimal temporally connected labellings of tree graphs contain exactly $2n-3$ labels and exactly 2 labels on all edges except one, independently of whether one tries to minimise or maximise the density. For proofs of these minimisation costs, we refer the reader to [18] for the original proofs in the gossiping context, or to Theorem 2 in [22] and Corollary 3 in [29] in the temporal graph context.

For trees, the results are mixed: on one hand we exactly determined both maximum temporal cost and maximum temporality of tree graphs $T^+(\mathtt{Trees})$ and $\tau^+(\mathtt{Trees})$, but on the other hand both are very sparse and do not improve upon lower bounds of maximum temporal cost T^+ and maximum temporality τ^+.

4 Cycle graphs

We focus here on finding dense labellings of cycle graphs, which we know exist thanks to the parity construction. For this, we decided to adapt temporal graph generator STGen from [7] (the description and adaptation of which is omitted in this version), which ultimately leads us to the densest labelling in this paper, the generator labelling. The main idea is to distribute even labels to an arbitrary edge, odd labels to the incident non-labelled edges, and so forth for the next non-labelled edges switching between even and odd labels with some additional complex rules.

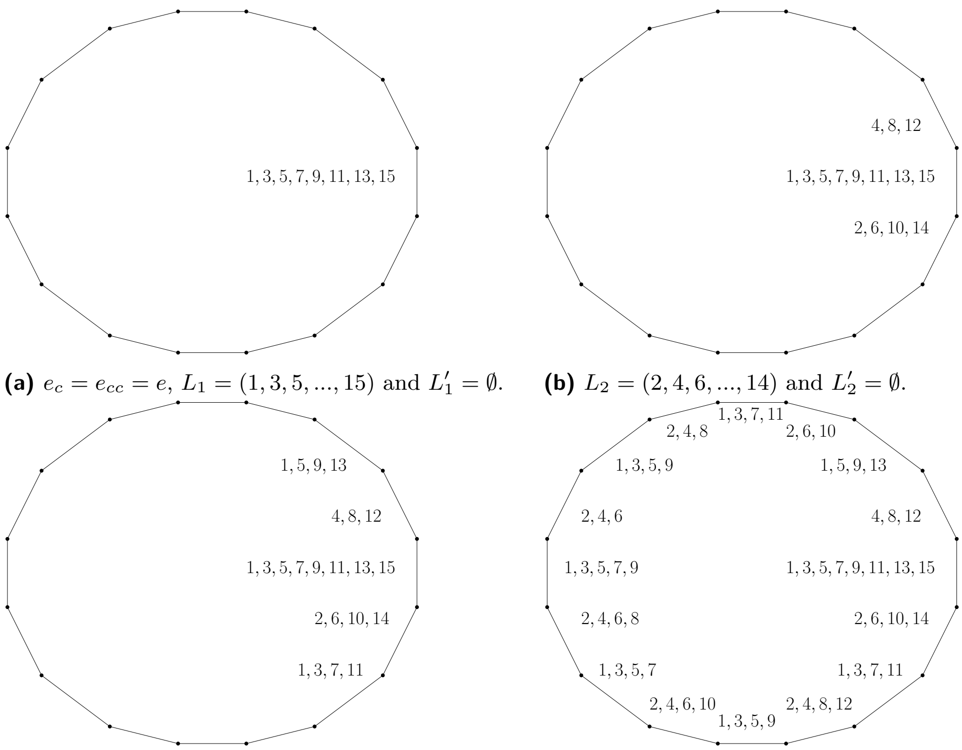
Proving this labelling results in a minimal and temporally connected graph for any order n cycle graph is complex due to the inherent unreadability of the multiple journeys in these temporal graphs. For this reason, we introduce a different type of representation which is very similar to the so-called link stream representation used in various temporal graph theory papers [25, 26, 34], and thus by slight misuse of terminology, we simply refer to it as the link stream representation. Link streams intuitively focus more on the time edge aspect of a temporal graph, and less on the structure of the underlying graph. Since we already know the underlying structure of the graph (being a cycle graph), link streams are perfect to represent our generator labelling. As an illustrative example, the link stream representation of Figure 4 is given in Figure 5, where all edges of the cycle are represented in one dimension, the horizontal dimension, and time is represented in the other, the vertical dimension. A "label" is thus represented as a time edge at the intersection of the corresponding edge and time value.

In this representation, a journey informally corresponds to a (possibly steep) "staircase" of time edges which, obeying the flow of time, cannot go down. We now remind and define some concepts concerning journeys in this setting, all of which are illustrated in Figure 5. A prefix of a journey $(v_1, \ell_1, v_2, \ell_2, ..., \ell_{k-1}, v_k)$ is a part of the journey cut back from the arrival vertex, *i.e.* $(v_1, \ell_1, v_2, \ell_2, ..., \ell_{i-1}, v_i)$ for some $i \leq k$, and a suffix of a journey is a part

Algorithm 1 Generator labelling (see also Figure 4).

---

input : even cycle graph G of order n, edge e of G
output : temporal graph $\mathcal{G} = (G, \lambda)$ with generator labelling λ
$L, L_1 \leftarrow$ `oddNumbersBetween`$(1, n-1)$ `/* ascending order */`
$L_2 \leftarrow$ `evenNumbersBetween`$(1, n-1)$ `/* ascending order */`
$L', L'_1, L'_2 \leftarrow \emptyset$
$e_c, e_{cc} \leftarrow e$
while $e_c = e$ *or* $e_c \neq e_{cc}$ **do**
  `addLabels`(L', e_c)
  `addLabels`(L', e_{cc})
  $e' \leftarrow e_c$
  for ℓ *in* L **do**
    `addLabel`(ℓ, e')
    if $e' = e_c$ **then** $e' \leftarrow e_{cc}$ **else** $e' \leftarrow e_c$
  `moveSmallestLabelFromTo`(L, L')
  `removeLargestLabelFrom`(L)
  $e_c \leftarrow$ `nextClockwiseEdge`(e_c)
  $e_{cc} \leftarrow$ `nextCounterClockwiseEdge`(e_{cc})
  if $L = L_1$ **then** $L \leftarrow L_2$ **else** $L \leftarrow L_1$
  if $L' = L'_1$ **then** $L' \leftarrow L'_2$ **else** $L' \leftarrow L'_1$
`addLabel`(`largestLabel`$(e_c) + 2, e_c$)



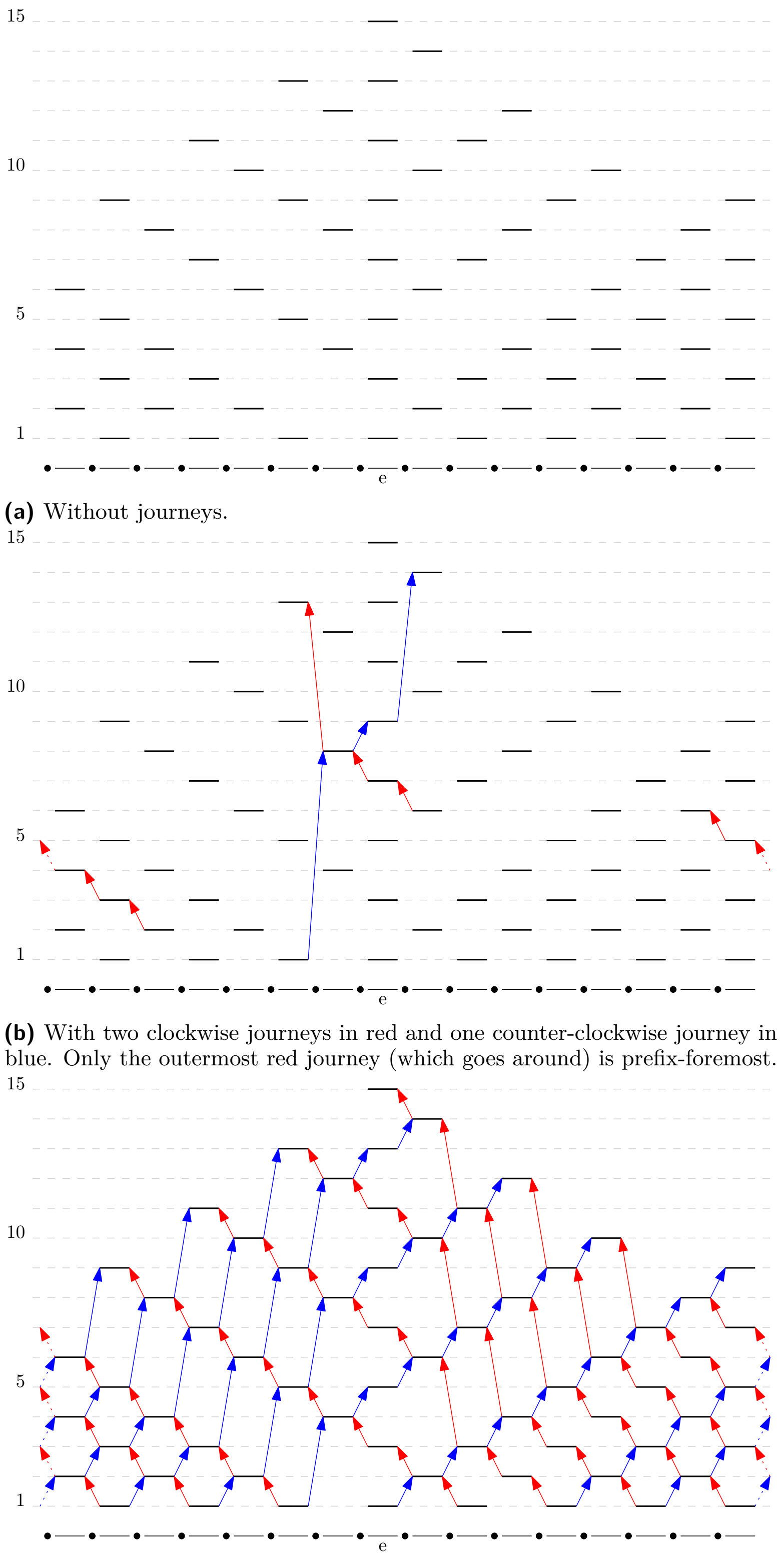
(a) $e_c = e_{cc} = e$, $L_1 = (1, 3, 5, ..., 15)$ and $L'_1 = \emptyset$.

(b) $L_2 = (2, 4, 6, ..., 14)$ and $L'_2 = \emptyset$.

(c) $L_1 = (3, 5, 7, 9, 13)$ and $L'_1 = (1)$.

(d) $L_1 = \emptyset$, $L'_1 = (1, 3, 5, 7)$ and $e_c = e_{cc}$ again.

Figure 4 The generator labelling for $n = 16$, starting on rightmost edge e, and repeating `while` loop with lists in captions. After the `while` loop finishes, label 9 is added on the last (leftmost) edge.



(a) Without journeys.

(b) With two clockwise journeys in red and one counter-clockwise journey in blue. Only the outermost red journey (which goes around) is prefix-foremost.

(c) With all dominating journeys.

Figure 5 The generator labelling for $n = 16$ in the link stream representation, with edge e shown in the middle. The rightmost edge connects the outermost vertices, allowing journeys to go around.

of the journey cut back from the starting vertex, *i.e.* $(v_j, \ell_j, v_{j+1}, \ell_{j+1}, ..., \ell_{k-1}, v_k)$ again for some $j \leq k$. A foremost journey from vertex u to vertex v is a journey which arrives at the earliest time among all journeys from u to v. A prefix-foremost journey from u to v is a journey which prefixes are all foremost, In other words, a prefix-foremost journey always takes the earliest edges possible on its journey. We define a clockwise journey as a journey which takes only time edges to the left, and a counter-clockwise journey is composed of only time-edges going to the right. Finally, we define dominating journeys as clockwise (*resp.* counter-clockwise) journeys such that no other clockwise (*resp.* counter-clockwise) journey exists which covers all its vertices or more.

We show in a very technical proof by induction that the generator labelling essentially "grows" domination journeys and creates new ones, as it is applied to larger and larger cycles graphs, and that these dominating journeys are necessary and ensure temporal connectivity.

▶ **Theorem 6** ($\star$)**.** *The generator labelling yields a minimal temporally connected graph.*

The temporal cost of the generator labelling is the largest presented in this paper, and with it also comes the largest temporality, namely on edge e. In the full version of this work, we show that the generator labelling works on all even cycles, and also give an adaptation for odd cycles. Analysing both gives us the main results.

▶ **Theorem 7.** $T^+(\mathtt{Cycles}) \geq \frac{1}{4}n^2 + \frac{3}{4}$.

▶ **Theorem 8.** $\tau^+(\mathtt{Cycles}) \geq \lceil \frac{1}{2}n \rceil$.

5 Conclusion

In conclusion, we provided some general upper bounds and proved tight or lower bound results for some basic graph classes on how dense a labelling they can admit. Also, our proposed generator labelling beats the previously densest labellings, in both temporal cost and temporality.

Since we allow a labelling to assign no labels to any edge, a density result for class $\mathtt{C}$ translates as a lower bound for any class $\mathtt{C}'$ if for all graphs $G' \in \mathtt{C}'$, there exists $G \in C$ such that G is an edge-deleted subgraph of G'. Conversely, a density result for class $\mathtt{C}$ implies an upper bound for any superclasses of $\mathtt{C}$. Together, this means that our (tight and lower bound) results for `Trees`, `Cycles`, and `Cacti`, thus respectively transfer for `Connected`, the class of connected graphs, `Hamiltonian`, the class of Hamiltonian graphs, and `Circumference` c, the class of graphs of circumference c.

We omitted in this version (again, see full version on arXiv [13]) the section on cactus graphs, where essentially the pivot labelling is used to gather information towards a largest cycle in the graph, instead of towards a pivot vertex, then apply the generator labelling on that cycle, and finally broadcast information outwards again with the pivot labelling, leading to the density results presented in Figure 1.

In terms of future work, one clear option is to consider other types of labellings. There are already some lower bounds known, such as Axiotis and Fotakis' construction from [2] for happy labellings, although we beat this construction slightly by adapting the ad-hoc labelling to become a happy labelling (omitted in this version). A lower bound for strict labellings comes from [23], in which Klobas et al. label all edges of an odd cycle with all integers up to $\lfloor \frac{n}{2} \rfloor$; another labelling attaining the same lower bound is the complete graph with label 1 on all edges. Another direction for future work is considering computational complexity of associated problems. The corresponding minimisation problems are all polynomial-time solvable, with the exception of deciding whether the minimum temporality of a graph is 1, which is NP-complete [16]. Our results only prove polynomial-time solvability for trees, and containment in APX for cycles and cacti of large circumference.

References

1 Eleni C Akrida, Leszek Gąsieniec, George B Mertzios, and Paul G Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61(3), 2017.

2 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 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

3 Brenda Baker and Robert Shostak. Gossips and telephones. *Discrete Mathematics*, 2:191–193, 1972.

4 Stefan Balev, Yoann Pigné, Eric Sanlaville, and Jason Schoeters. Temporally connected components. *Available at SSRN 4590651*, 2023.

5 Davide Bilò, Gianlorenzo D'Angelo, Luciano Gualà, Stefano Leucci, and Mirko Rossi. Sparse temporal spanners with low stretch. In *30th Annual European Symposium on Algorithms (ESA 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

6 Davide Bilò, Gianlorenzo D'Angelo, Luciano Gualà, Stefano Leucci, and Mirko Rossi. Blackout-tolerant temporal spanners. In *International Symposium on Algorithms and Experiments for Wireless Sensor Networks*, pages 31–44. Springer, 2022.

7 Arnaud Casteigts. Efficient generation of simple temporal graphs up to isomorphism. *Github repository*, 2020. URL: `https://github.com/acasteigts/STGen`.

8 Arnaud Casteigts, Timothée Corsini, and Writika Sarkar. Simple, strict, proper, happy: A study of reachability in temporal graphs. *Theoretical Computer Science*, page 114434, 2024.

9 Arnaud Casteigts, Kitty Meeks, George B. Mertzios, and Rolf Niedermeier. Temporal Graphs: Structure, Algorithms, Applications (Dagstuhl Seminar 21171). *Dagstuhl Reports*, 11(3):16–46, 2021. `doi:10.4230/DagRep.11.3.16`.

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

11 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.

12 Daniel Charbonneau, Benjamin Blonder, and Anna Dornhaus. Social insects: a model system for network dynamics. *Temporal networks*, pages 217–244, 2013.

13 Esteban Christiann, Eric Sanlaville, and Jason Schoeters. On inefficiently connecting temporal networks, 2023. `arXiv:2312.07117`.

14 Nicholas S DiBrita, Khouloud Eledlebi, Hanno Hildmann, Lucas Culley, and Abdel F Isakovic. Temporal graphs and temporal network characteristics for bio-inspired networks during optimization. *Applied Sciences*, 12(3):1315, 2022.

15 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.

16 F. Göbel, J.Orestes Cerdeira, and H.J. Veldman. Label-Connected Graphs and the Gossip Problem. *Discrete Mathematics*, 87(1):29–40, January 1991. `doi:10.1016/0012-365X(91)90068-D`.

17 András Hajnal, Eric C Milner, and Endre Szemerédi. A cure for the telephone disease. *Canadian Mathematical Bulletin*, 15(3):447–450, 1972.

18 Hovhannes Harutyunyan, Arthur Liestman, Joseph Peters, and Dana Richards. Broadcasting and Gossiping. In Ping Zhang, editor, *Handbook of Graph Theory, Second Edition*, volume

20134658, pages 1477–1494. Chapman and Hall/CRC, December 2013. Series Title: Discrete Mathematics and Its Applications. doi:10.1201/b16132-87.

19 Danny Hermelin, Yuval Itzhaki, Hendrik Molter, and Rolf Niedermeier. Temporal unit interval independent sets. In *1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

20 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.

21 Arindam Khanda, Federico Corò, Francesco Betti Sorbelli, Cristina M Pinotti, and Sajal K Das. Efficient route selection for drone-based delivery under time-varying dynamics. In *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pages 437–445. IEEE, 2021.

22 Nina Klobas, George B Mertzios, Hendrik Molter, and Paul G Spirakis. The complexity of computing optimum labelings for temporal connectivity. In *47th International Symposium on Mathematical Foundations of Computer Science*, 2022.

23 Nina Klobas, George B Mertzios, Hendrik Molter, and Paul G Spirakis. Realizing temporal graphs from fastest travel times. *arXiv preprint arXiv:2302.08860*, 2023.

24 David C Kutner and Laura Larios-Jones. Temporal reachability dominating sets: contagion in temporal graphs. *arXiv preprint arXiv:2306.06999*, 2023.

25 Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8:1–29, 2018.

26 Jundong Li, Kewei Cheng, Liang Wu, and Huan Liu. Streaming link prediction on dynamic attributed networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 369–377, 2018.

27 Fuqiang Liu, Jingbo Tian, Luis Miranda-Moreno, and Lijun Sun. Adversarial danger identification on temporally dynamic graphs. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

28 Andrea Marino and Ana Silva. Coloring temporal graphs. *Journal of Computer and System Sciences*, 123:171–185, 2022.

29 George B Mertzios, Othon Michail, and Paul G Spirakis. Temporal network optimization subject to connectivity constraints. *Algorithmica*, 81(4):1416–1449, 2019.

30 George B Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. Computing maximum matchings in temporal graphs. In *37th International Symposium on Theoretical Aspects of Computer Science*, 2020.

31 U Rajeshwari and B Sathish Babu. Real-time credit card fraud detection using streaming analytics. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 439–444. IEEE, 2016.

32 Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, S Yu Philip, Lifang He, and Bo Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

33 Mathilde Vernet, Yoann Pigné, and Eric Sanlaville. A study of connectivity on dynamic graphs: computing persistent connected components. *4OR*, pages 1–29, 2022.

34 Peixiang Zhao, Charu Aggarwal, and Gewen He. Link prediction in graph streams. In *2016 IEEE 32nd international conference on data engineering (ICDE)*, pages 553–564. IEEE, 2016.

A Proof(s) of Section 2

▶ **Theorem 2.** *The maximum temporal cost $T^+ \leq n^2 - n - 1$ and the maximum temporality $\tau^+ \leq n - 1$.*

Proof. By Lemma 1, a minimal temporally connected graph equals the union of any of its n distinct-root spanning temporal branchings.

Concerning maximum temporal cost, the worst-case scenario for the total number of labels in such a graph is when these temporal branchings are all disjoint. This results in a labelling using $n-1$ labels for each branching (as they are spanning), of which there are n, resulting in a total of n^2-n labels.

Consider however the smallest label ℓ^- used in the graph, say on edge $e=\{u,v\}$. This label can only be part of the spanning temporal branching of root u, denoted $\mathcal{B}_u$, or of the spanning temporal branching of root v, denoted $\mathcal{B}_v$, since it's unreachable from any other vertex. Suppose *w.l.o.g.* ℓ^- is part of $\mathcal{B}_u$. We know v must reach u through some journey in $\mathcal{B}_v$ arriving at some time ℓ. Note that ℓ can be removed from $\mathcal{B}_v$, and ℓ^- added. Indeed, for all w, any journey $v \rightsquigarrow w$ in $\mathcal{B}_v$ is either maintained by the swap, or passes through u earlier with ℓ^-, meaning $\mathcal{B}_v$ remains a spanning temporal branching. Thus, label ℓ^- can be considered part of two spanning temporal branchings, decrementing the total amount of labels to n^2-n-1.

Concerning maximum temporality, the worst-case number of labels on an edge in such a graph is when the spanning temporal branchings are all label-disjoint and all use one same edge $e=\{u,v\}$, resulting in an edge having 1 label for each branching, of which there are n, resulting in a total of n labels.

Note however that the label from the branching corresponding to root u, and the label from branching corresponding to root v, are necessarily the same label, since otherwise the later of the two would be redundant, as both branchings can use the earlier label. Thus edge e would have $n-1$ labels. ◀

B Proof(s) of Section 3

We first need the following lemma concerning dense path graphs.

▶ **Lemma 9.** $T^+(\mathtt{Paths}) = 2n-3$

Proof. The pivot labelling and Lemma 3 apply to path graphs, implying that $2n-3 \leq T^+(\mathtt{Paths}) \leq 2n-2$.

Suppose w.l.o.g. the vertices of a path graph to be, from one leaf vertex to the other, v_1, v_2 etc. up to v_n. Note that for a path graph to be temporally connected, we only need to ensure that both extremities, v_1 and v_n, can reach each other, via a journey in one direction, and a journey in the other. Indeed, any other pair of vertices can use these journeys to reach each other. Thanks to this, we can reason on temporally connected path graphs and only need to worry about the reachability between these two leaf vertices, instead of between all vertices.

Suppose by contradiction that a minimal temporally connected path graph G exists with $2n-2$ labels. By our previous argument, we have that any label which is not part of either the journey from v_1 to v_n, or of the journey from v_n to v_1, is redundant. Each of these two journeys is composed of $n-1$ labels, meaning that to obtain $2n-2$ necessary labels from only these two journeys, we must have that all labels on the two journeys must be distinct. There must exist one edge e_i such that the corresponding labels have the smallest difference among all edges of path G. By the temporal nature of journeys, there cannot be more than one such edge as the difference must necessarily increase on edges further away from e_i. Consider the labels on edge e_i and incident edges e_{i-1} and e_{i+1}, denoted $\ell_i^{\rightarrow}$, $\ell_i^{\leftarrow}$, $\ell_{i-1}^{\rightarrow}$, $\ell_{i-1}^{\leftarrow}$, $\ell_{i+1}^{\rightarrow}$, and $\ell_{i+1}^{\leftarrow}$. (If edge e_i only has one incident edge, then ignore the following concerning the non-existent other edge and labels.) Consider labels $\ell_{i-1}^{\rightarrow} < \ell_i^{\rightarrow} < \ell_{i+1}^{\rightarrow}$ to be part of $v_1 \rightsquigarrow v_n$, and $\ell_{i-1}^{\leftarrow} > \ell_i^{\leftarrow} > \ell_{i+1}^{\leftarrow}$ to be part of $v_n \rightsquigarrow v_1$. Suppose *w.l.o.g.* that $\ell_i^{\rightarrow} > \ell_i^{\leftarrow}$. Thus we have that $\ell^{\rightarrow}_{i+1} > \ell^{\rightarrow}_{i} > \ell^{\leftarrow}_{i}$ and also $\ell^{\leftarrow}_{i+1} < \ell^{\leftarrow}_{i} < \ell^{\rightarrow}_{i}$. On edge e_{i-1} however, two cases are possible:

- $\ell^{\rightarrow}_{i-1} < \ell^{\leftarrow}_{i}$: this means label $\ell^{\rightarrow}_{i}$ is redundant as $\ell^{\rightarrow}_{i-1} < \ell^{\leftarrow}_{i} < \ell^{\rightarrow}_{i+1}$;
- $\ell^{\leftarrow}_{i-1} > \ell^{\rightarrow}_{i}$: this means label $\ell^{\leftarrow}_{i}$ is redundant as $\ell^{\leftarrow}_{i+1} < \ell^{\rightarrow}_{i} < \ell^{\leftarrow}_{i-1}$.

Due to the inequalities presented, and the fact that the difference between $\ell^{\leftarrow}_{i-1}$ and $\ell^{\rightarrow}_{i-1}$ must be larger than the difference between $\ell^{\leftarrow}_{i}$ and $\ell^{\rightarrow}_{i}$, at least one of the previous cases must be present, meaning at least one label must be redundant which is a contradiction. ◀

▶ **Theorem 5.** *$T^+(\texttt{Trees}) = 2n - 3$.*

Proof. Since path graphs are tree graphs, Lemma 9 gives an upper bound of $2n - 3$, which is attained by the pivot labelling. ◀

C Proof(s) of Section 4

The following technical lemmas are needed to then prove minimality of the generator labelling.

▶ **Lemma 10.** *In a cycle graph $G = (V, E)$ without any journey covering V, a clockwise journey $\mathcal{J} = ((\{v_j, v_{j-1}\}, t_1), (\{v_{j-1}, v_{j-2}\}, t_2), ..., (\{v_i, v_{i-1}\}, t_k))$ is dominating if and only if:*

- *it starts at the earliest date possible, i.e. there exists no time edge $(\{v_j, v_{j-1}\}, t)$ nor $(\{v_{j+1}, v_j\}, t)$ with $t < t_1$;*
- *it ends at the latest date possible, i.e. there exists no time edge $(\{v_i, v_{i-1}\}, t)$ nor $(\{v_{i-1}, v_{i-2}\}, t)$ with $t > t_k$;*
- *no other time edges exist between successive time edges, i.e. for all successive pairs of time edges $(\{v_a, v_{a-1}\}, t_b)$ and $(\{v_{a-1}, v_{a-2}\}, t_c > t_b)$ of $\mathcal{J}$, there exists no time edge $(\{v_a, v_{a-1}\}, t)$ or $(\{v_{a-1}, v_{a-2}\}, t)$ with $t_b < t < t_c$.*

A symmetric characterisation holds for counter-clockwise journeys.

Proof. Let us focus on clockwise journeys, the proof being symmetric for counter-clockwise journeys. Suppose by contradiction that a journey $\mathcal{J}$ obeys the three criteria, but is not dominating, meaning there exists some other distinct clockwise journey $\mathcal{J}'$ covering the same vertex set (or more). A case analysis follows depending on which vertex $\mathcal{J}'$ starts.

If journey $\mathcal{J}'$ starts from any vertex that $\mathcal{J}$ covers, except for v_j, then to ensure $\mathcal{J}'$ covers the vertices of $\mathcal{J}$, it needs to go all the way around the cycle graph and thus cover V, which is explicitly excluded in this lemma.

If $\mathcal{J}'$ starts at vertex v_j, and it contains some earlier time edge than the corresponding time edge in $\mathcal{J}$, then $\mathcal{J}$ doesn't respect criterion three (as this earlier time edge exists between successive time edges of $\mathcal{J}$). If instead it contains a later time edge, then $\mathcal{J}'$ must rejoin or cross $\mathcal{J}$ at some point (since $\mathcal{J}$ uses the latest date of edge $\{v_i, v_{i-1}\}$ by criterion two), implying $\mathcal{J}$ again does not respect criterion three. Of course, if $\mathcal{J}'$ does not contain any earlier or later time edge than $\mathcal{J}$, then it will end in the same manner as $\mathcal{J}$ without any way of continuing by criterion two, meaning it is identical to $\mathcal{J}$.

Lastly, if $\mathcal{J}'$ starts on any other vertex, then since $\mathcal{J}$ respects criterion one, $\mathcal{J}'$ must arrive later than t_1 on edge $\{v_j, v_{j-1}\}$. By the same argument as before concerning $\mathcal{J}'$ having a later time edge than $\mathcal{J}$, the former must rejoin or cross the latter at some point, implying $\mathcal{J}$ does not respect criterion three.

Since all cases end in some contradiction, being either $\mathcal{J}$ breaking one of the criteria or $\mathcal{J}'$ being identical to $\mathcal{J}$, we can thus conclude that $\mathcal{J}$ is dominating. ◀

▶ **Lemma 11.** *In a cycle graph $G = (V, E)$ without any journey covering V, a pair of clockwise and counter-clockwise journeys is necessary if:*
- *both start at some same vertex v;*
- *both are prefix-foremost;*
- *both are a suffix of a dominating journey;*
- *and they do not cross (except on vertex v).*

Proof. We prove that such a pair of clockwise and counter-clockwise journeys, say journey $\mathcal{J}_v^w$ which clockwise goes up to vertex w, and journey $\mathcal{J}_v^u$ which counter-clockwise goes up to vertex u, is necessary for reachability from v to w and from v to u respectively. *W.l.o.g.* we give the proof for the former only, the proof for the latter being symmetric. We first prove that no counter-clockwise journey can reach vertex w, and then that the only clockwise journey that can reach w is journey $\mathcal{J}_v^w$, from which it follows that this journey is necessary.

First note that vertex v cannot reach further than u in a counter-clockwise manner. Indeed, if by contradiction we suppose there is some counter-clockwise journey $\mathcal{J}_v^{u'}$ from v to vertex u' such that u' is positioned further than u, then *w.l.o.g.* we may consider $\mathcal{J}_v^{u'}$ to be prefix-foremost (if it is not, then we can make it so by changing its time edges for the earliest possible). Since $\mathcal{J}_v^u$ and $\mathcal{J}_v^{u'}$ are both prefix-foremost clockwise journeys, we know that $\mathcal{J}_v^u$ must be a prefix of journey $\mathcal{J}_v^{u'}$, *i.e.* $\mathcal{J}_v^{u'}$ is the concatenation of journeys $\mathcal{J}_v^u$ and say $\mathcal{J}_u^{u'}$. Journey $\mathcal{J}_v^u$ is a suffix of a dominating journey $\mathcal{J}_d$, meaning no other counter-clockwise journey covers the vertices of $\mathcal{J}_d$ or more, but now we obtain our contradiction: the concatenation of $\mathcal{J}_d$ and $\mathcal{J}_u^{u'}$ covers more vertices (the only case where this wouldn't be true is if $\mathcal{J}_d$ covered V which is explicitly excluded from the lemma statement). Since $\mathcal{J}_v^w$ and $\mathcal{J}_v^u$ don't cross (except on vertex v), we now know that v cannot reach w through a counter-clockwise journey.

To finish the proof, we show $\mathcal{J}_v^w$ is the only clockwise journey that can reach w, meaning all its edges are necessary. Suppose by contradiction another clockwise journey $\mathcal{J}$ exists from v to w. It cannot be a prefix-foremost journey, as by definition this would be journey $\mathcal{J}_v^w$. Since $\mathcal{J}$ is not prefix-foremost, it uses some edge e with label l' whereas $\mathcal{J}_v^w$ uses edge e with some label $l < l'$. However, we remind the reader that $\mathcal{J}_v^w$ is a suffix of a dominating journey $\mathcal{J}_d$. Altogether, this means another journey exists covering the same vertices as $\mathcal{J}_d$, being the concatenation of the prefix of $\mathcal{J}_d$ up to vertex v, and $\mathcal{J}$. By definition of dominating journeys, this is a contradiction. ◀

We note that if the pair of journeys from Lemma 11 collectively covers V, then vertex v can reach all vertices through these journeys.

▶ **Theorem 6.** *The generator labelling yields a minimal temporally connected graph.*

Proof. The proof is by induction. Consider cycle graph C_8 as our base case. Apply the generator labelling starting on some edge e. Let e be composed of vertices v_{-2} and v_{-1}, and let vertices v_i be the vertices in the clockwise direction of e, with i the (clockwise) hop distance between e and v_i, and similarly, let vertices $v_{-(i+2)}$ be the vertices in the counter-clockwise direction of e, with i the (counter-clockwise) hop distance between e and $v_{-(i+2)}$. Compute the dominating journeys, see Figure 6. Let us define some specific sets of time edges as follows. Let the five earliest time edges on edges $\{v_{-2}, v_{-1}\}$, $\{v_{-1}, v_1\}$ and $\{v_1, v_2\}$ be referred to as the seed, and the three latest time edges as the trunk. Let the latest two time edges on edges $\{v_{-4}, v_{-3}\}$ and $\{v_{-3}, v_{-2}\}$ be referred to as a branch, as well as the ones on edges $\{v_2, v_3\}$ and $\{v_3, v_4\}$. More specifically, let the former be branch B_2, as the dominating clockwise journey starting at vertex v_2 ends in these edges, and the latter

B_{-2} as the dominating counter-clockwise journey starting at vertex v_{-2} ends here. Finally, let the other time edges be referred to as the base. Note that all time edges are part of some dominating journey, and that all dominating journeys start on a time edge with time 1 in the base (except for the two dominating journeys starting in the seed) and that all dominating journeys end in the latest time edges of branches (except for two dominating journeys ending in the trunk).

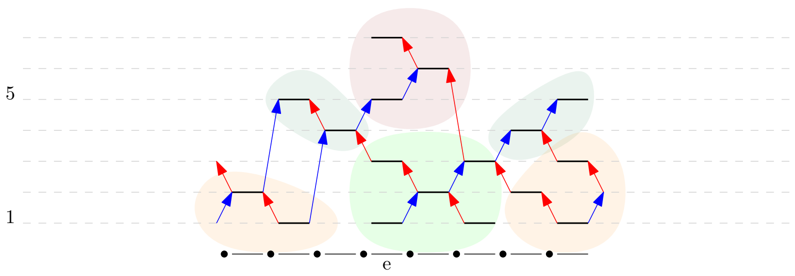
It is possible to claim minimality and temporal connectivity for this small temporal graph, although we specifically point out that Lemma 11 can be used on all vertices (except for those of the seed) to prove necessity of all dominating journeys starting on these vertices, and reachability of all these vertices. Note that now only the time edges of the base remain to be proven necessary. The time edges of the counter-clockwise dominating journey can be proven necessary by applying Lemma 11 on vertex v_{-2}, but it cannot be applied to vertex v_2 to prove the remaining time edges necessary, as its counter-clockwise prefix-foremost journey is not dominating. However, they are proven necessary through the ad hoc argument: without these edges v_2 cannot reach v_{-4}, as any clockwise journey by definition cannot reach it except for its clockwise dominating journey which relied on these time edges, and any counter-clockwise journey reaches at most vertex v_4. Concerning reachability of the vertices of the seed, it is possible for them to use the dominating journeys starting in the seed to go to any other vertex (note that these journeys do not cross outside of in the seed, but do cover V).

Now, in the inductive step, this structure of seed, base, trunk and branches remains or gets extended when growing the generator labelling for some C_n to some C_{n+4}. More precisely, the seed remains as is, the base gets extended with so-called roots, the trunk with a so-called apex, and the branches with leaves. Also, two new branches are created in every inductive step, which sprout from the top of the trunk. Underlying all this, we prove that in the inductive step, the dominating journeys get extended slightly, are modified, or created, in a precise manner which ultimately allows us to again use Lemma 11 to prove minimality and temporal connectivity, in a very similar manner as how we did for C_8.

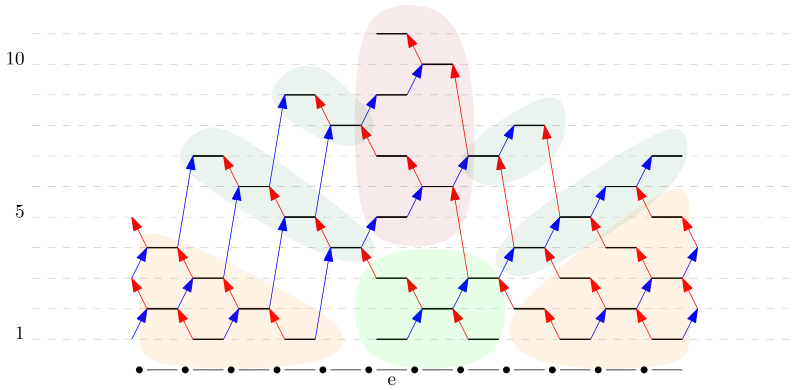
Now, suppose we have a cycle graph C_{4k} with the generator labelling which has been proven minimal and temporally connected, specifically through applying Lemma 11 on all vertices except for the seed. Add vertices v_{2k+1}, v_{2k+2}, v_{-2k-1}, and v_{-2k-2} and the corresponding edges to the link stream representation so as to obtain cycle C_{4k+4}. See Figure 6. This effectively breaks dominating journeys which previously used edge $\{v_{-2k}, v_{2k}\}$, whose time edges now belong to edge $\{v_{-2k}, v_{-2k-1}\}$. We will patch these halves of dominating journeys back together in what follows, although not exactly with their original half. Note that now the generator labelling for C_{4k+4} is exactly this labelling, with some additional time edges which are all later, *i.e.* for all additional time edges (e, t), there exists no already present time edge $(e, t' > t)$. Let the three additional time edges extending the trunk be referred to as the apex, let the leaves be the pairs of additional time edges extending the branches (as well as creating branches B_{2k} and B_{-2k} from the trunk), and let the remaining additional time edges be the roots, which extend the base.

Let us start by proving these additional edges are all part of some dominating journey.

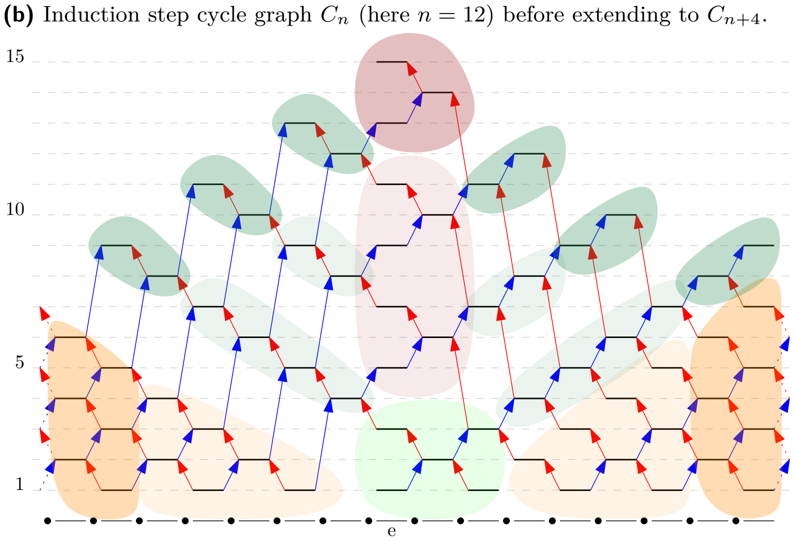
Leaves extending branches B_i extend the corresponding dominating journey starting from vertex v_i. The three conditions from Lemma 10 hold for this extended journey, as it still starts at time 1, no additional time edges have been added in between the time edges it uses, and it ends at the latest time possible at the top of its respective leaves. Regarding the leaves that create a new branch B_{2k} and B_{-2k}, these extend dominating journeys from vertices v_{2k} and v_{-2k} which ended at the top of the trunk before (we can observe two of these exist



(a) Base case cycle graph C_8, with seed, base, trunk and two branches.



(b) Induction step cycle graph C_n (here $n = 12$) before extending to C_{n+4}.



(c) Apex, leaves, and roots extending C_n to obtain C_{n+4}, adding two branches.

Figure 6 Illustration of the proof by induction for Theorem 6, with the seed (light green), trunk (brown), branches (green) and base (light brown). At the induction step, the apex, leaves, and roots are shown in the same but less transparent colour as the structures they extend.

in C_8, and below we prove that in every inductive step two new such journeys are created). These extended journeys remain dominating by the same argument as for other branches. All leaves are thus part of a dominating journey. Note that this extends (by exactly two time edges) basically half of all previously existing dominating journeys.

The other half of previous dominating journeys are broken up through the addition of the four new vertices and edges. The roots serve to patch these journeys back together. Note that before, all these dominating journeys started in the base, cycled around, and climbed through the branches to finish at the top of some branch. More specifically, such a dominating journey starting from a vertex v_i finished at the top of branch B_{i-1} for $i > 0$ and at the top of branch B_{i+1} otherwise (an exception being the journey starting from v_{-3} which ends at the second largest time edge of the branch B_{-2}). We show this remains true after the inductive step. The reconstructed dominating journey, suppose from vertex v_i for $i > 0$ (the explanation being symmetric for $i < 0$), starts of with the same time edges it had before in the base until it reaches the roots. This means this part of the journey respects two of the conditions of Lemma 10, being it starts at time 1, and no time edges exist in between its time edges as this journey was dominating before and the additional time edges are all later. Now the earliest four time edges possible are taken to continue this journey in the roots, cycling around to the other side of the link stream. This also respects the condition of having no time edges in between these four time edges, as the roots are densely packed by definition. The journey is now four time steps too late to reconnect with the other half it had before, connect it instead with the half of the journey which previously started from vertex v_{i-4} for which it arrived through the roots perfectly on time. This latter half also respects the condition of having no time edges in between its time edges due to part of a dominating journey before, and no additional time edges have been added in between these time edges. Since our journey now follows the part of the dominating journey which previously started at v_{i-4}, it arrives at branch B_{i-5}, but can now continue through the leaves of B_{i-3} and finally B_{i-1} to end at the latest edge. By construction, this continuation through the leaves respects the conditions of Lemma 10 since there are no time edges in between, and it ends at the latest time possible. There are two exceptions to this: the reconstruction of the dominating journey starting from vertex v_3 only uses three time edges from the roots, before directly ending in the leaves of branch B_2, and the one starting from vertex v_5 directly goes up through the leaves of B_2 and B_4 after the roots. Both reconstructed journeys are dominating as well. Note that now all dominating journeys starting at vertices v_i with $-2k \leq i \leq 2k$ have been extended (the ones cycling around have been broken apart and refitted first but in terms of length have been extended as well) by exactly two time edges compared to their previous length in C_{4k}.

Observe that some of the earliest roots have not been shown to be part of a dominating journey yet, and also that some halves of previous dominating journeys have not been refitted together yet. We show another four dominating journeys exist which start from the four new vertices, use these earliest roots, as well as the remaining parts of previous dominating journeys, and two of these journeys use the time edges of the apex. Proving these four journeys are dominating is done through again applying Lemma 10 with the arguments already explained for the other dominating journeys, and thus we decide to forgo doing this again four more times. The dominating journey starting at v_{-2k-1} goes clockwise, starting at time 1 and the four earliest roots, then it continues with the part of the previous dominating journey ending in branch B_{-2k+4}, and goes up through the leaves of branches B_{-2k+2} and B_{-2k}, finishing at the latest time edge of the latter. Starting at vertex v_{-2k-2}, we have a counter-clockwise dominating journey, starting at time 1 using only one root, linking up with part of a previous dominating journey which finished at branch B_{2k-2}, which is extended further through branch B_{2k} and the apex to end on the second latest time edge. We note that the last two dominating journeys do not cross, except on the first edge, and cover V. Continuing, we have a clockwise dominating journey starting at v_{2k+2} and time 1, using two roots before using part of a previous dominating journey leading up to branch B_{-2k-2}, continuing through the leaves of B_{-2k} and ending in the apex on the largest time edge. Finally, there is a counter-clockwise dominating journey from vertex v_{2k+1} using three roots cycling around the link stream, pairing up with part of a previous dominating journey leading up to branch B_{2k-4} which then continues through leaves of B_{2k+2} and B_{2k} to end on the largest time edge of that branch. Again, these two journeys do not cross, except on the first edge, and cover V.

Thus, we have that all time edges are part of a dominating journey, and that basically the same journeys from C_{4k} remain in C_{4k+4} (albeit some of them recombined differently) and were extended by exactly two time edges. Since for all vertices but those of the seed, Lemma 11 was used to prove necessity of the corresponding dominating journeys, this lemma can be used again for these vertices to prove necessity of their corresponding dominating journeys, as well as reachability of these vertices to all others. For the time edges and vertices of the seed, the argument used for C_8 can be generalized to prove necessity and reachability as well. Finally, the last four dominating journeys which start on the four new vertices, can use Lemma 11 as well, since their clockwise and counter-clockwise prefix-foremost journeys are dominating and can collectively cover V.

In conclusion, we have proven that the base case, being the generator labelling for C_8, is minimal and temporally connected. Then, for any inductive step from C_{4k} to C_{4k+4}, minimality and temporal connectivity are conserved in this labelling. Thus, the generator labelling produces a minimal and temporally connected graph for any size $4k$. (The generator labelling works for C_4 as well, it is easy to check, but the structure of the labelling was easier to explain with C_8, as for C_4 the labelling is composed of only the seed.) ◀