Temporal Network Optimization Subject to Connectivity Constraints*

George B. Mertzios[1], Othon Michail[2],
Ioannis Chatzigiannakis[2], and Paul G. Spirakis[2,3]

[1] School of Engineering and Computing Sciences, Durham University, UK
[2] Computer Technology Institute & Press "Diophantus" (CTI), Patras, Greece
[3] Department of Computer Science, University of Liverpool, UK
Email: george.mertzios@durham.ac.uk, {michailo, ichatz, spirakis}@cti.gr

Abstract. In this work we consider *temporal networks*, i.e. networks defined by a *labeling* λ assigning to each edge of an *underlying graph* G a set of *discrete* time-labels. The labels of an edge, which are natural numbers, indicate the discrete time moments at which the edge is available. We focus on *path problems* of temporal networks. In particular, we consider *time-respecting* paths, i.e. paths whose edges are assigned by λ a strictly increasing sequence of labels. We begin by giving two efficient algorithms for computing shortest time-respecting paths on a temporal network. We then prove that there is a *natural analogue of Menger's theorem* holding for arbitrary temporal networks. Finally, we propose two *cost minimization parameters* for temporal network design. One is the *temporality* of G, in which the goal is to minimize the maximum number of labels of an edge, and the other is the *temporal cost* of G, in which the goal is to minimize the total number of labels used. Optimization of these parameters is performed subject to some *connectivity constraint*. We prove several lower and upper bounds for the temporality and the temporal cost of some very basic graph families such as rings, directed acyclic graphs, and trees.

1 Introduction

A *temporal* (or *dynamic*) *network* is, loosely speaking, a network that changes with time. This notion encloses a great variety of both modern and traditional networks such as information and communication networks, social networks, transportation networks, and several physical systems.

In this work, embarking from the foundational work of Kempe *et al.* [KKK00], we consider *discrete time*, that is, we consider networks in which changes occur at discrete moments in time, e.g. days. This choice is not only a very natural

* Supported in part by (i) the project FOCUS implemented under the "ARISTEIA" Action of the OP "Education and Lifelong Learning" and co-funded by the EU (ESF) and Greek National Resources, (ii) the FET EU IP project MULTIPLEX under contract no 317532, and (iii) the EPSRC Grant EP/G043434/1. Full version: http://ru1.cti.gr/aigaion/?page=publication&kind=single&ID=977

abstraction of many real systems but also gives to the resulting models a purely combinatorial flavor. In particular, we consider those networks that can be described via an underlying graph G and a labeling λ assigning to each edge of G a (possibly empty) set of discrete labels. Note that this is a generalization of the single-label-per-edge model used in [KKK00], as we allow many time-labels to appear on an edge. These labels are drawn from the natural numbers and indicate the discrete moments in time at which the corresponding connection is available. For example, in the case of a communication network, availability of a communication link at some time t may mean that a communication protocol is allowed to transmit a data packet over that link at time t.

In this work, we initiate the study of the following fundamental network design problem: "*Given an underlying (di)graph G, assign labels to the edges of G so that the resulting temporal graph $\lambda(G)$ minimizes some parameter while satisfying some connectivity property*". In particular, we consider two cost optimization parameters for a given graph G. The first one, called *temporality* of G, measures the maximum number of labels that an edge of G has been assigned. The second one, called *temporal cost* of G, measures the total number of labels that have been assigned to all edges of G (i.e. if $|\lambda(e)|$ denotes the number of labels assigned to edge e, we are interested in $\sum_{e\in E}|\lambda(e)|$). Each of these two cost measures can be minimized subject to some particular connectivity property $\mathcal{P}$ that the temporal graph $\lambda(G)$ has to satisfy. In this work, we consider two very basic connectivity properties. The first one, that we call the *all paths* property, requires the temporal graph to preserve every simple path of its underlying graph, where by "preserve a path of G" we mean that the labeling should provide at least one strictly increasing sequence of labels on the edges of that path (we also call such a path *time-respecting*).

For an illustration, consider a directed ring $u_1, u_2, \ldots, u_n$. We want to determine the temporality of the ring subject to the *all paths* property, that is, we want to find a labeling λ that preserves every simple path of the ring and at the same time minimizes the maximum number of labels of an edge. Consider the paths $P_1 = (u_1, \ldots, u_n)$ and $P_2 = (u_{n-1}, u_n, u_1, u_2)$. It is immediate to observe that an increasing sequence of labels on the edges of path P_1 implies a decreasing pair of labels on edges (u_{n-1}, u_n) and (u_1, u_2). On the other hand, path P_2 uses first (u_{n-1}, u_n) and then (u_1, u_2) thus it requires an increasing pair of labels on these edges. It follows that in order to preserve both P_1 and P_2 we have to use a second label on at least one of these two edges, thus the temporality is at least 2. Next, consider the labeling that assigns to each edge (u_i, u_{i+1}) the labels $\{i, n+i\}$, where $1 \leq i \leq n$ and $u_{n+1} = u_1$. It is not hard to see that this labeling preserves all simple paths of the ring. Since the maximum number of labels that it assigns to an edge is 2, we conclude that the temporality is also at most 2. In summary, the temporality of preserving all simple paths of a directed ring is 2.

The other connectivity property that we define, called the *reach* property, requires the temporal graph to preserve a path from node u to node v whenever v is reachable from u in the underlying graph. Furthermore, the minimization of each of our two cost measures can be affected by some problem-specific constraints on the labels that we are allowed to use. We consider here one of the most natural constraints, namely an upper bound of the *age* of the constructed labeling λ, where the age of a labeling λ is defined to be equal to the maximum label of λ minus its minimum label plus 1. Now the goal is to minimize the cost parameter, e.g. the temporality, satisfy the connectivity property, e.g. *all paths*, and additionally guarantee that the age does not exceed some given natural k. Returning to the ring example, it is not hard to see, that if we additionally restrict the age to be at most $n-1$ then we can no longer preserve all paths of a ring using at most 2 labels per edge. In fact, we must now necessarily use the worst possible number of labels, i.e. $n-1$ on every edge.

Minimizing such parameters may be crucial as, in most real networks, making a connection available and maintaining its availability does not come for free. At the same time, such a study is important from a purely graph-theoretic perspective as it gives some first insight into the structure of specific families of temporal graphs (e.g. *no temporal ring exists with fewer than* $n+1$ *labels*). Finally, we believe that our results are a first step towards answering the following fundamental question: "*To what extent can algorithmic and structural results of graph theory be carried over to temporal graphs?*". For example, is there an analogue of Menger's theorem for temporal graphs? One of the results of the present work is an affirmative answer to the latter question.

1.1 Related Work

Single-label Temporal Graphs and Menger's Theorem. The model of temporal graphs that we consider in this work is a direct extension of the single-label model studied in [Ber96] and [KKK00] to allow for many labels per edge. In [KKK00], Kempe *et al.*, among other things, proved that there is no analogue of Menger's theorem, at least in its original formulation, for arbitrary single-label temporal networks. In this work, we go a step ahead showing that if one reformulates Menger's theorem in a way that takes time into acount then a very natural temporal analogue of Menger's theorem is obtained. Furthermore, in the present work, we consider a path as *time-respecting* if its edges have strictly increasing labels and not non-decreasing as in the above papers.
Continuous Availabilities (Intervals). Some authors have naturally assumed that an edge may be available for continuous time-intervals. The techniques used there are quite different than those needed in the discrete case [XFJ03, FT98].
Distributed Computing on Dynamic Networks. A notable set of recent works has studied (distributed) computation in *worst-case* dynamic networks in which the topology may change arbitrarily from round to round (see e.g. [KLO10, MCS12]). Population protocols [AAD$^+$06] and variants [MCS11a] are collections of passively mobile finite-state agents that compute something useful in the limit. Another interesting direction assumes random network dynamicity and the interest is on determining "good" properties of the dynamic network that hold with high probability and on designing protocols for distributed tasks [CMM$^+$08, AKL08]. For introductory texts cf. [CFQS12, MCS11b, Sch02].

Distance Labeling. A distance labeling of a graph G is an assignment of unique labels to the vertices of G so that the distance between any two vertices can be inferred from their labels alone [GPPR01, KKKP04]. There, the labeling parameter to be minimized is the binary length of an appropriate distance encoding, which is different from our cost parameters.

1.2 Contribution

In Section 2, we formally define the model of temporal graphs under consideration and provide all further necessary definitions. In Section 3, we give two efficient algorithms for computing shortest time-respecting paths. Then in Section 4 we present an analogue of Menger's theorem which we prove valid for arbitrary temporal graphs. In the full paper, we also apply our Menger's analogue to substantially simplify the proof of a recent result on distributed token gathering. In Section 5, we formally define the temporality and temporal cost optimization metrics for temporal graphs. In Section 5.1, we provide several upper and lower bounds for the temporality of some fundamental graph families such as rings, directed acyclic graphs (DAGs), and trees, as well as an interesting trade-off between the temporality and the age of rings. Furthermore, we provide in Section 5.2 a generic method for computing a lower bound of the temporality of an arbitrary graph w.r.t. the *all paths* property, and we illustrate its usefulness in cliques and planar graphs. Finally, we consider in Section 5.3 the temporal cost of a digraph G w.r.t. the *reach* property, when additionally the age of the resulting labeling $\lambda(G)$ is restricted to be the smallest possible. We prove that this problem is **APX**-hard. To prove our claim, we first prove (which may be of interest in its own right) that the Max-XOR(3) problem is **APX**-hard via a PTAS reduction from Max-XOR. In Max-XOR(3) problem, we are given a 2-CNF formula ϕ, every literal of which appears in at most 3 clauses, and we want to compute the greatest number of clauses of ϕ that can be simultaneously XOR-satisfied. Then we provide a PTAS reduction from Max-XOR(3) to our temporal cost minimization problem. On the positive side, we provide an $(r(G)/n)$-factor approximation algorithm for the latter problem, where $r(G)$ denotes the total number of reachabilities in G.

2 Preliminaries

Given a (di)graph $G = (V, E)$, a *labeling* of G is a mapping $\lambda : E \to 2^{\mathbb{N}}$, that is, a labeling assigns to each edge of G a (possibly empty) set of natural numbers, called *labels*.

Definition 1. *Let G be a (di)graph and λ be a labeling of G. Then $\lambda(G)$ is the* temporal graph *(or* dynamic graph*) of G with respect to λ. Furthermore, G is the* underlying graph *of $\lambda(G)$.*

We denote by $\lambda(E)$ the multiset of all labels assigned to the underlying graph by the labeling λ and by $|\lambda| = |\lambda(E)|$ their cardinality (i.e. $|\lambda| = \sum_{e \in E} |\lambda(e)|$).

We also denote by $\lambda_{\min} = \min\{l \in \lambda(E)\}$ the minimum label and by $\lambda_{\max} = \max\{l \in \lambda(E)\}$ the maximum label assigned by λ. We define the *age* of a temporal graph $\lambda(G)$ as $\alpha(\lambda) = \lambda_{\max} - \lambda_{\min} + 1$. Note that in case $\lambda_{\min} = 1$ then we have $\alpha(\lambda) = \lambda_{\max}$. For every graph G we denote by $\mathcal{L}_G$ the set of all possible labelings λ of G. Furthermore, for every $k \in \mathbb{N}$, we define $\mathcal{L}_{G,k} = \{\lambda \in \mathcal{L}_G : \alpha(\lambda) \leq k\}$.

For every time $r \in \mathbb{N}$, we define the *rth instance of a temporal graph* $\lambda(G)$ as the static graph $\lambda(G, r) = (V, E(r))$, where $E(r) = \{e \in E : r \in \lambda(e)\}$ is the (possibly empty) set of all edges of the underlying graph G that are assigned label r by labeling λ. A temporal graph $\lambda(G)$ may be also viewed as a *sequence of static graphs* $(G_1, G_2, \ldots, G_{\alpha(\lambda)})$, where $G_i = \lambda(G, \lambda_{\min} + i - 1)$ for all $1 \leq i \leq \alpha(\lambda)$. Another, often convenient, representation of a temporal graph is the following.

Definition 2. *The* static expansion *of a temporal graph $\lambda(G)$ is a DAG $H = (S, A)$ defined as follows. If $V = \{u_1, u_2, \ldots, u_n\}$ then $S = \{u_{ij} : \lambda_{\min} - 1 \leq i \leq \lambda_{\max}, 1 \leq j \leq n\}$ and $A = \{(u_{(i-1)j}, u_{ij'}) : \text{if } j = j' \text{ or } (u_j, u'_j) \in E(i) \text{ for some } \lambda_{\min} \leq i \leq \lambda_{\max}\}$.*

A *journey* (or *time-respecting path*) J of a temporal graph $\lambda(G)$ is a path $(e_1, e_2, \ldots, e_k)$ of the underlying graph $G = (V, E)$, where $e_i \in E$, together with labels $l_1 < l_2 < \ldots < l_k$ such that $l_i \in \lambda(e_i)$ for all $1 \leq i \leq k$. In words, a journey is a path that uses strictly increasing edge-labels. If labeling λ defines a journey on some path P of G then we also say that λ *preserves* P. A natural notation for a journey is $(e_1, l_1), (e_2, l_2), \ldots, (e_k, l_k)$ where each (e_i, l_i) is called a *time-edge*. A (u, v)-journey J is called *foremost from time* $t \in \mathbb{N}$ if $l_1 \geq t$ and l_k is minimized. We say that a journey J *leaves from node u* (*arrives at*, resp.) *at time t* if (u, v, t) ((v, u, t), resp.) is a time-edge of J. Two journeys are called *out-disjoint* (*in-disjoint*, respectively) if they never leave from (arrive at, resp.) the same node at the same time. If, in addition to the labeling λ, a positive weight $w(e) > 0$ is assigned to every edge $e \in E$, then we get a *weighted* temporal graph. If this is the case, then a journey J is called *shortest* if it minimizes the sum of the weights of its edges.

Throughout the text, unless otherwise stated, we denote by n the number of nodes of (di)graphs and by $d(G)$ the diameter of a (di)graph G, that is the length of the longest shortest path between any two nodes of G. Finally, by δ_u we denote the degree of a node $u \in V(G)$ (in case of an undirected graph G).

3 Journey Problems

Theorem 1. *Let $\lambda(G)$ be a* temporal graph*, $s \in V$ be a source node, and t_{start} a time s.t. $\lambda_{\min} \leq t_{start} \leq \lambda_{\max}$. There is an algorithm that correctly computes for all $w \in V \backslash \{s\}$ a foremost (s, w)-journey from time t_{start}. The running time of the algorithm is $O(n\alpha^3(\lambda) + |\lambda|)$.*

Theorem 2. *Let $\lambda(G)$ be a weighted temporal graph and let $s, t \in V$. Assume also that $|\lambda(e)| = 1$ for all $e \in E$. Then, we can compute a shortest journey J between s and t in $\lambda(G)$ (or report that no such journey exists) in $O(m \log m + \sum_{v \in V} \delta_v^2) = O(n^3)$ time, where $m = |E|$.*

4 A Menger's Analogue for Temporal Graphs

In this section, we prove that, in contrast to an important negative result from [KKK00], there is a natural analogue of Menger's theorem that is valid for all temporal networks. In the full paper, we also apply our theorem to substantially simplify the proof of a recent token gathering result.

When we say that we remove *node departure time* (u, t) we mean that we remove *all edges leaving* u *at time* t, i.e. we remove the set $\{(u, v) \in E : t \in \lambda(u, v)\}$. So, when we ask how many node departure times are needed to separate two nodes s and v we mean how many node departure times must be selected so that after the removal of all the corresponding time-edges the resulting temporal graph has no (s, v)-journey.

Theorem 3 (Menger's Temporal Analogue). *Take any temporal graph $\lambda(G)$, where $G = (V, E)$, with two distinguished nodes s and v. The maximum number of out-disjoint journeys from s to v is equal to the minimum number of node departure times needed to separate s from v.*

Proof. Assume, in order to simplify notation, that $\lambda_{\min} = 1$. Take the static expansion $H = (S, A)$ of $\lambda(G)$. Let $\{u_{i1}\}$ and $\{u_{in}\}$ represent s and v over time, respectively (first and last colums, respectively), where $0 \leq i \leq \lambda_{\max}$. We extend H as follows. For each u_{ij}, $0 \leq i \leq \lambda_{\max} - 1$, with at least 2 outgoing edges to nodes different than $u_{(i+1)j}$, e.g. to nodes $u_{(i+1)j_1}, u_{(i+1)j_2}, \ldots, u_{(i+1)j_k}$, we add a new node w_{ij} and the edges (u_{ij}, w_{ij}) and $(w_{ij}, u_{(i+1)j_1})$, $(w_{ij}, u_{(i+1)j_2}), \ldots, (w_{ij}, u_{(i+1)j_k})$. We also define an edge capacity function $c : A \to \{1, \lambda_{\max}\}$ as follows. All edges of the form $(u_{ij}, u_{(i+1)j})$ take capacity $\lambda_{\max}$ and all other edges take capacity 1. We are interested in the maximum flow from u_{01} to $u_{\lambda_{\max} n}$. As this is simply a usual static flow network, the max-flow min-cut theorem applies stating that the maximum flow from u_{01} to $u_{\lambda_{\max} n}$ is equal to the minimum of the capacity of a cut separating u_{01} from $u_{\lambda_{\max} n}$. Finally, observe that (i) the maximum number of out-disjoint journeys from s to v is equal to the maximum flow from u_{01} to $u_{\lambda_{\max} n}$ and (ii) the minimum number of node departure times needed to separate s from v is equal to the minimum of the capacity of a cut separating u_{01} from $u_{\lambda_{\max} n}$. $\square$

5 Minimum Cost Temporal Connectivity

In this section, we introduce (in Definition 3) the *temporality* and *temporal cost* measures. These measures can be minimized subject to some particular connectivity property $\mathcal{P}$ that the labeled graph $\lambda(G)$ has to satisfy. For simplicity of notation, we consider the connectivity property $\mathcal{P}$ as a subset of the set $\mathcal{L}_G$ of all possible labelings λ on the (di)graph G. Furthermore, the minimization of each of these two cost measures can be affected by some problem-specific constraints on the labels that we are allowed to use. We consider one of the most natural constraints, namely an upper bound on the *age* of the constructed labeling.

Definition 3. *Let $G = (V, E)$ be a (di)graph, $\alpha_{\max} \in \mathbb{N}$, and $\mathcal{P}$ be a connectivity property. Then the* temporality *of $(G, \mathcal{P}, \alpha_{\max})$ is*

$$\tau(G, \mathcal{P}, \alpha_{\max}) = \min_{\lambda \in \mathcal{P} \cap \mathcal{L}_{G,\alpha_{\max}}} \max_{e \in E} |\lambda(e)|$$

and the temporal cost *of $(G, \mathcal{P}, \alpha_{\max})$ is*

$$\kappa(G, \mathcal{P}, \alpha_{\max}) = \min_{\lambda \in \mathcal{P} \cap \mathcal{L}_{G,\alpha_{\max}}} \sum_{e \in E} |\lambda(e)|$$

Furthermore $\tau(G, \mathcal{P}) = \tau(G, \mathcal{P}, \infty)$ and $\kappa(G, \mathcal{P}) = \kappa(G, \mathcal{P}, \infty)$.

Note that Definition 3 can be stated for an arbitrary property $\mathcal{P}$ of the labeled graph $\lambda(G)$ (e.g. some proper coloring-preserving property). Nevertheless, we only consider here $\mathcal{P}$ to be a connectivity property of $\lambda(G)$. In particular, we investigate the following two connectivity properties $\mathcal{P}$:

- *all-paths*$(G) = \{\lambda \in \mathcal{L}_G$: for all simple paths P of G, λ preserves $P\}$,
- *reach*$(G) = \{\lambda \in \mathcal{L}_G$: for all $u, v \in V$ where v is reachable from v in G, λ preserves at least one simple path from u to $v\}$.

5.1 Basic Properties of Temporality Parameters

5.1.1 Preserving All Paths. We begin with some simple observations on $\tau(G, \textit{all paths})$. Recall that given a (di)graph G our goal is to label G so that all simple paths of G are preserved by using as few labels per edge as possible. First note that if $p(G)$ is the length of the longest path in G then $\tau(G, \text{all paths}) \leq p(G)$ for all graphs G: just give to every edge the labels $\{1, 2, \ldots, p(G)\}$.

A topological sort of a digraph G is a linear ordering of its nodes such that if G contains an edge (u, v) then u appears before v in the ordering. It is well known that a digraph G can be topologically sorted iff is a DAG.

Proposition 1. *If G is a DAG then $\tau(G, \textit{all paths}) = 1$.*

Proof. Take a topological sort $u_1, u_2, \ldots, u_n$ of G. Give to every edge (u_i, u_j), where $i < j$, label i. □

5.1.2 Preserving All Reachabilities. Now, instead of preserving all paths, we impose the apparently simpler requirement of preserving just a single path between every reachability pair $u, v \in V$. We claim that it is sufficient to understand how $\tau(G, \textit{reach})$, behaves on strongly connected digraphs. Let $\mathcal{C}(G)$ be the set of all strongly connected components of a digraph G. The following lemma proves that, w.r.t. the *reach* property, the temporality of any digraph G is upper bounded by the maximum temporality of its components.

Lemma 1. $\tau(G, \textit{reach}) \leq \max_{C \in \mathcal{C}(G)} \tau(C, \textit{reach})$ *for every digraph G.*

Lemma 1 implies that any upper bound on the temporality of preserving the reachabilities of strongly connected digraphs can be used as an upper bound on the temporality of preserving the reachabilities of general digraphs. An interesting question is whether there is some bound on $\tau(G, reach)$ either for all digraphs or for specific families of digraphs. By using Lemma 1, it can be proved that indeed there is a very satisfactory generic upper bound.

Theorem 4. *$\tau(G, reach) \leq 2$ for all digraphs G.*

5.1.3 Restricting the Age. Now notice that for all G we have $\tau(G, reach, d(G)) \leq d(G)$; recall that $d(G)$ denotes the diameter of (di)graph G. Indeed it suffices to label each edge by $\{1, 2, \ldots, d(G)\}$. Thus, a clique G has trivially $\tau(G, reach, d(G)) = 1$ as $d(G) = 1$ and we can only have large $\tau(G, reach, d(G))$ in graphs with large diameter. For example, a directed ring G of size n has $\tau(G, reach, d(G)) = n - 1$. Indeed, assume that from some edge e, label $1 \leq i \leq n-1$ is missing. It is easy to see that there is some shortest path between two nodes of the ring that in order to arrive by time $n-1$ must use edge e at time i. As this label is missing, it uses label $i+1$, thus it arrives by time n which is greater than the diameter. On a ring we can preserve the diameter only if all edges have the labels $\{1, 2, \ldots, n-1\}$.

On the other hand, there are graphs with large diameter in which $\tau(G, reach, d(G))$ is small. This may also be the case even if G is strongly connected. For example, consider the graph with nodes $u_1, u_2, \ldots, u_n$ and edges (u_i, u_{i+1}) and (u_{i+1}, u_i) for all $1 \leq i \leq n-1$. In words, we have a directed line from u_1 to u_n and an inverse one from u_n to u_1. The diameter here is $n-1$ (e.g. the shortest path from u_1 to u_n) but $\tau(G, reach, d(G)) = 1$: simply label one path $1, 2, ..., n-1$ and label the inverse one $1, 2, ..., n-1$ again, i.e. give to edges (u_i, u_{i+1}) and (u_{n-i+1}, u_{n-i+2}) label i. Now consider an undirected tree T.

Theorem 5. *If T is an undirected tree then $\tau(T, \textit{all paths}, d(T)) \leq 2$.*

We next present an interesting trade-off between the temporality and the age of a directed ring.

Theorem 6. *If G is a directed ring and $\alpha = (n-1)+k$, where $1 \leq k \leq n-1$, then $\tau(G, \textit{all paths}, \alpha) = \Theta(n/k)$ and in particular $\lfloor \frac{n-1}{k+1} \rfloor + 1 \leq \tau(G, \textit{all paths}, \alpha) \leq \lceil \frac{n}{k+1} \rceil + 1$. Moreover, $\tau(G, \textit{all paths}, n-1) = n-1$ (i.e. when $k = 0$).*

5.2 A Generic Method for Lower Bounding Temporality

We show here that there are graphs G for which $\tau(G, \textit{all paths}) = \Omega(p(G))$ (recall that $p(G)$ denotes the length of the longest path in G), that is graphs in which the optimum labeling, w.r.t. temporality, is very close to the trivial labeling $\lambda(e) = \{1, 2, \ldots, p(G)\}$, for all $e \in E$.

Definition 4. *Call a set $K = \{e_1, e_2, \ldots, e_k\} \subseteq E(G)$ of edges of a digraph G an* edge-kernel *if for every permutation $\pi = (e_{i_1}, e_{i_2}, \ldots, e_{i_k})$ of K there is a simple path of G that visits all edges of K in the ordering defined by π.*

The following theorem states that an edge-kernel of size k needs at least k labels on some edge(s).

Theorem 7 (Edge-kernel Lower Bound). *If a digraph G contains an edge-kernel of size k then $\tau(G, \textit{all paths}) \geq k$.*

The usefulness of Theorem 7 is that it allows us to establish a lower bound k on the temporality of a graph G by only proving the existence of an edge-kernel of size k in G. We now apply this to complete digraphs and planar graphs.

Lemma 2. *If G is a complete digraph of order n then it has an edge-kernel of size $\lfloor n/2 \rfloor$.*

Now Theorem 7 implies that if G is a complete digraph then $\lfloor n/2 \rfloor \leq \tau(G, \text{all paths}) \leq n - 1$.

Lemma 3. *There exist planar graphs having edge-kernels of size $\Omega(n^{\frac{1}{3}})$.*

5.3 Computing the Cost

5.3.1 Hardness of Approximation. Consider a boolean formula ϕ in conjunctive normal form with two literals in every clause (2-CNF). Let τ be a truth assignment of the variables of ϕ and $\alpha = (\ell_1 \vee \ell_2)$ be a clause of ϕ. Then α is *XOR-satisfied* in τ, if one of the literals $\{\ell_1, \ell_2\}$ of the clause α is true in τ and the other one is false in τ. The number of clauses of ϕ that are XOR-satisfied in τ is denoted by $|t(\phi)|$. The formula ϕ is *XOR-satisfiable* if there exists a truth assignment τ of ϕ such that every clause of ϕ is XOR-satisfied in τ. The *Max-XOR* problem is the following maximization problem: given a 2-CNF formula ϕ, compute the greatest number of clauses of ϕ that can be simultaneously XOR-satisfied in a truth assignment τ, i.e. compute the greatest value for $|t(\phi)|$. The *Max-XOR(k)* problem is the special case of the the Max-XOR problem, where every literal of the input formula ϕ appears in at most k clauses of ϕ. Max-XOR is known to be **APX**-hard, i.e. it does not admit a PTAS unless **P** = **NP** [KMSV99, CKS01]. In the next lemma we prove that Max-XOR(3) remains **APX**-hard by providing a PTAS reduction from Max-XOR.

Lemma 4. *The Max-XOR(3) problem is* **APX***-hard.*

Now we provide a reduction from the Max-XOR(3) problem to the problem of computing $\kappa(G, \textit{reach}, d(G))$. Let ϕ be an instance formula of Max-XOR(3) with n variables $x_1, x_2, \ldots, x_n$ and m clauses. Since every variable x_i appears in ϕ (either as x_i or as $\overline{x_i}$) in at most 3 clauses, it follows that $m \leq \frac{3}{2}n$. We will construct from ϕ a graph G_ϕ having length of a directed cycle at most 2. Then, as we prove in Theorem 8, $\kappa(G_\phi, \textit{reach}, d(G_\phi)) \leq 39n - 4m - 2k$ if and only if there exists a truth assignment τ of ϕ with $|t(\phi)| \geq k$, i.e. τ XOR-satisfies at least k clauses of ϕ. Since ϕ is an instance of Max-XOR(3), we can replace every clause $(\overline{x_i} \vee \overline{x_j})$ by the clause $(x_i \vee x_j)$ in ϕ, since $(\overline{x_i} \vee \overline{x_j}) = (x_i \vee x_j)$ in XOR. Furthermore, whenever $(\overline{x_i} \vee x_j)$ is a clause of ϕ, where $i < j$, we can replace this clause by $(x_i \vee \overline{x_j})$, since $(\overline{x_i} \vee x_j) = (x_i \vee \overline{x_j})$ in XOR. Thus, we can assume w.l.o.g. that every clause of ϕ is either of the form $(x_i \vee x_j)$ or $(x_i \vee \overline{x_j})$, $i < j$.

For every $i = 1, 2, \ldots, n$ we construct the graph $G_{\phi,i}$ of Figure 1. Note that the diameter of $G_{\phi,i}$ is $d(G_{\phi,i}) = 9$ and the maximum length of a directed cycle in $G_{\phi,i}$ is 2. In this figure, we call the induced subgraph of $G_{\phi,i}$ on the 13 vertices $\{s^{x_i}, u_1^{x_i}, \ldots, u_6^{x_i}, v_1^{x_i}, \ldots, v_6^{x_i}\}$ the *trunk* of $G_{\phi,i}$. Furthermore, for every $p \in \{1, 2, 3\}$, we call the induced subgraph of $G_{\phi,i}$ on the 5 vertices $\{u_{7,p}^{x_i}, u_{8,p}^{x_i}, v_{7,p}^{x_i}, v_{8,p}^{x_i}, t_p^{x_i}, \}$ the *pth branch* of $G_{\phi,i}$. Finally, we call the edges $u_6^{x_i} u_{7,p}^{x_i}$ and $v_6^{x_i} v_{7,p}^{x_i}$ the *transition edges* of the pth branch of $G_{\phi,i}$. Furthermore, for every $i = 1, 2, \ldots, n$, let $r_i \leq 3$ be the number of clauses in which variable x_i appears in ϕ. For every $1 \leq p \leq r_i$, we assign the pth appearance of the variable x_i (either as x_i or as $\overline{x_i}$) in a clause of ϕ to the pth branch of $G_{\phi,i}$.

Consider now a clause $\alpha = (\ell_i \vee \ell_j)$ of ϕ, where $i < j$. Then, by our assumptions on ϕ, it follows that $\ell_i = x_i$ and $\ell_j \in \{x_j, \overline{x_j}\}$. Assume that the literal ℓ_i (resp. ℓ_j) of the clause α corresponds to the pth (resp. to the qth) appearance of the variable x_i (resp. x_j) in ϕ. Then we identify the vertices of the pth branch of $G_{\phi,i}$ with the vertices of the qth branch of $G_{\phi,j}$ as follows. If $\ell_j = x_j$ then we identify the vertices $u_{7,p}^{x_i}, u_{8,p}^{x_i}, v_{7,p}^{x_i}, v_{8,p}^{x_i}, t_p^{x_i}$ with the vertices $v_{7,q}^{x_j}, v_{8,q}^{x_j}, u_{7,q}^{x_j}, u_{8,q}^{x_j}, t_q^{x_j}$, respectively. Otherwise, if $\ell_j = \overline{x_j}$ then we identify the vertices $u_{7,p}^{x_i}, u_{8,p}^{x_i}, v_{7,p}^{x_i}, v_{8,p}^{x_i}, t_p^{x_i}$ with the vertices $u_{7,q}^{x_j}, u_{8,q}^{x_j}, v_{7,q}^{x_j}, v_{8,q}^{x_j}, t_q^{x_j}$, respectively. This completes the construction of the graph G_ϕ. Note that, similarly to the graphs $G_{\phi,i}$, $1 \leq i \leq n$, the diameter of G_ϕ is $d(G_\phi) = 9$ and the maximum length of a directed cycle in G_ϕ is 2. Furthermore, note that for each of the m clauses of ϕ, one branch of a gadget $G_{\phi,i}$ coincides with one branch of a gadget $G_{\phi,j}$, where $1 \leq i < j \leq n$, while every $G_{\phi,i}$ has three branches. Therefore G_ϕ has exactly $3n - 2m$ branches which belong to only one gadget $G_{\phi,i}$, and m branches that belong to two gadgets $G_{\phi,i}, G_{\phi,j}$.

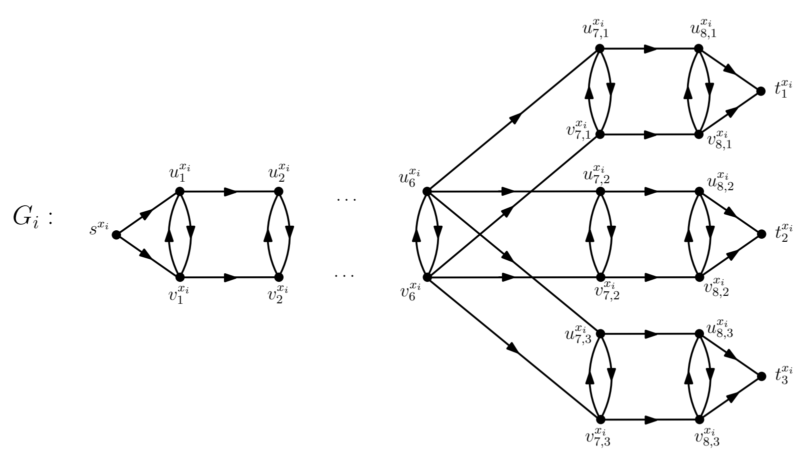

Fig. 1. The gadget $G_{\phi,i}$ for the variable x_i.

Theorem 8. *There exists a truth assignment τ of ϕ with $|t(\phi)| \geq k$ if and only if $\kappa(G_\phi, reach, d(G_\phi)) \leq 39n - 4m - 2k$.*

Using Theorem 8, we are now ready to prove the main theorem of this section.

Theorem 9 (Hardness of Approximating the Temporal Cost). *The problem of computing $\kappa(G, reach, d(G))$ is* **APX***-hard, even when the maximum length of a directed cycle in G is 2.*

Proof. Denote now by $\mathrm{OPT}_{\text{Max-XOR(3)}}(\phi)$ the greatest number of clauses that can be simultaneously XOR-satisfied by a truth assignment of ϕ. Then Theorem 8 implies that

$$\kappa(G_\phi, reach, d(G_\phi)) \leq 39n - 4m - 2 \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi)$$

Note that a random assignment XOR-satisfies each clause of ϕ with probability $\frac{1}{2}$, and thus we can easily compute (even deterministically) an assignment τ that XOR-satisfies $\frac{m}{2}$ clauses of ϕ. Therefore $\mathrm{OPT}_{\text{Max-XOR(3)}}(\phi) \geq \frac{m}{2}$, and thus, since every variable x_i appears in at least one clause of ϕ, it follows that $n \leq m \leq 2 \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi_1)$.

Assume that there is a PTAS for computing $\kappa(G, reach, d(G))$. Then, for every $\varepsilon > 0$ we can compute in polynomial time a labeling λ for the graph G_ϕ, such that $|\lambda| \leq (1+\varepsilon) \cdot \kappa(G_\phi, reach, d(G_\phi))$.

Given such a labeling λ we can compute by the sufficiency part ($\Leftarrow$) of the proof of Theorem 8 a truth assignment τ of ϕ such that $39n - 4m - 2|t(\phi)| \leq |\lambda|$, i.e. $2|t(\phi)| \geq 39n - 4m - |\lambda|$.

Therefore it follows by all the above that $2|t(\phi)| \geq 39n - 4m - (1+\varepsilon) \cdot \kappa(G_\phi, reach, d(G_\phi)) \geq 39n - 4m - (1+\varepsilon) \cdot \big(39n - 4m - 2 \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi)\big) = \varepsilon\,(4m - 39n) + 2(1+\varepsilon) \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi) \geq -35\varepsilon m + (2+2\varepsilon) \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi) \geq -35\varepsilon \cdot 2\mathrm{OPT}_{\text{Max-XOR(3)}}(\phi) + (2+2\varepsilon) \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi) = (2 - 68\varepsilon) \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi)$ and thus

$$|t(\phi)| \geq (1 - 34\varepsilon) \cdot \mathrm{OPT}_{\text{Max-XOR(3)}}(\phi).$$

That is, assuming a PTAS for computing $\kappa(G, reach, d(G))$, we obtain a PTAS for the Max-XOR(3) problem, which is a contradiction by Lemma 4. Therefore computing $\kappa(G, reach, d(G))$ is **APX**-hard. Finally, notice that the constructed graph G_ϕ has maximum length of a directed cycle at most 2. $\square$

5.3.2 Approximating the Cost. In this section, we provide an approximation algorithm for computing $\kappa(G, reach, d(G))$, which complements the hardness result of Theorem 9. Given a digraph G define, for every $u \in V$, u's reachability number $r(u) = |\{v \in V : v \text{ is reachable from } u\}|$ and $r(G) = \sum_{u \in V} r(u)$, that is $r(G)$ is the total number of reachabilities in G.

Theorem 10. *There is an $\frac{r(G)}{n-1}$-factor approximation algorithm for computing $\kappa(G, reach, d(G))$ on any weakly connected digraph G.*

References

[AAD+06] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, pages 235–253, March 2006.

[AKL08] C. Avin, M. Koucký, and Z. Lotker. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). In *Proceedings of the 35th international colloquium on Automata, Languages and Programming (ICALP), Part I*, pages 121–132. Springer-Verlag, 2008.

[Ber96] K. A. Berman. Vulnerability of scheduled networks and a generalization of Menger's theorem. *Networks*, 28[3]:125–134, 1996.

[CFQS12] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *IJPEDS*, 27[5]:387–408, 2012.

[CKS01] N. Creignou, S. Khanna, and M. Sudan. *Complexity classifications of boolean constraint satisfaction problems.* SIAM Monographs on Discrete Mathematics and Applications, 2001.

[CMM+08] A. E. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri. Flooding time in edge-markovian dynamic graphs. In *Proc. of the 27th ACM Symp. on Principles of distributed computing (PODC)*, pages 213–222, 2008.

[FT98] L. Fleischer and É. Tardos. Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23[3]:71–80, 1998.

[GPPR01] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance labeling in graphs. In *Proc. of the 12th annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 210–219, Philadelphia, PA, USA, 2001.

[KKK00] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. In *Proceedings of the 32nd annual ACM symposium on Theory of computing (STOC)*, pages 504–513, 2000.

[KKKP04] M. Katz, N. A. Katz, A. Korman, and D. Peleg. Labeling schemes for flow and connectivity. *SIAM Journal on Computing*, 34[1]:23–40, 2004.

[KLO10] F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proceedings of the 42nd ACM symposium on Theory of computing (STOC)*, pages 513–522, New York, NY, USA, 2010. ACM.

[KMSV99] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28[1]:164–191, 1999.

[MCS11a] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Mediated population protocols. *Theoretical Computer Science*, 412[22]:2434–2450, May 2011.

[MCS11b] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. *New Models for Population Protocols.* N. A. Lynch (Ed), Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool, 2011.

[MCS12] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Causality, influence, and computation in possibly disconnected synchronous dynamic networks. In *16th International Conference on Principles of Distributed Systems (OPODIS)*, pages 269–283, 2012.

[Sch02] C. Scheideler. Models and techniques for communication in dynamic networks. In *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 27–49, 2002.

[XFJ03] B. Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14[02]:267–285, 2003.