Connectivity and Inference Problems for Temporal Networks

David Kempe * Jon Kleinberg † Amit Kumar ‡

Abstract

Many network problems are based on fundamental relationships involving time. Consider, for example, the problems of modeling the flow of information through a distributed network, studying the spread of a disease through a population, or analyzing the reachability properties of an airline timetable. In such settings, a natural model is that of a graph in which each edge is annotated with a *time label* specifying the time at which its endpoints "communicated." We will call such a graph a *temporal network*. To model the notion that information in such a network "flows" only on paths whose labels respect the ordering of time, we call a path *time-respecting* if the time labels on its edges are non-decreasing. A model essentially equivalent to temporal networks was proposed recently by Berman, in work that raised a number of interesting open questions about their basic properties.

The central motivation for our work is the following question: how do the basic combinatorial and algorithmic properties of graphs change when we impose this additional temporal condition? The notion of a *path* is intrinsic to many of the most fundamental algorithmic problems on graphs; spanning trees, connectivity, flows, and cuts are some examples. When we focus on time-respecting paths in place of arbitrary paths, many of these problems acquire a character that is different from the traditional setting, but very rich in its own right. In focusing on these questions, we are additionally guided by issues that arise naturally in the study of broadcasting and gossiping in distributed networks.

We provide results on two types of problems for temporal networks. First, we consider *connectivity problems*, in which we seek disjoint time-respecting paths between pairs of nodes. The natural analogue of Menger's Theorem for node-disjoint paths fails in general for time-respecting paths; we give a non-trivial characterization of those graphs for which the theorem does hold in terms of an excluded subdivision theorem, and provide a polynomial-time algorithm for connectivity on this class of graphs. (The problem on general graphs is NP-complete.) We then define and study the class of *inference problems*, in which we seek to reconstruct a partially specified time labeling of a network in a manner consistent with an observed history of information flow.

1 Introduction

In a variety of settings, one encounters problems that are best modeled using a network with an explicit *time-ordering* on its edges. Although diverse in motivation, such problems involve a number of common themes, as indicated by the following examples.

Communication in distributed networks. As agents in a distributed network communicate over time, information flows in complex ways. Gossip protocols in distributed systems, for example, are based on the dissemination of information through a network using node-to-node transmissions [5, 9, 15, 16]. Suppose we are given a network in which nodes have been communicating for a period of time with their neighbors; each edge is labeled by the time(s) at which its endpoints exchanged information. Information "flows" along a path P in this network only if the time labels on the edges of P are monotonically non-decreasing; thus, such *time-respecting* paths are crucial structures in understanding the way in which information has disseminated through the network.

Epidemiology. The study of epidemics — the spread of infectious diseases — is a well-developed area of applied mathematics [1]. If we picture a network of individuals coming into contact with one another, there is a natural analogy to the previous setting; and indeed, this analogy

*Department of Computer Science, Cornell University, Ithaca NY 14853. Email: kempe@cs.cornell.edu.

†Department of Computer Science, Cornell University, Ithaca NY 14853. Email: kleinber@cs.cornell.edu. Supported in part by a David and Lucile Packard Foundation Fellowship, an Alfred P. Sloan Research Fellowship, an ONR Young Investigator Award, and NSF Faculty Early Career Development Award CCR-9701399.

‡Department of Computer Science, Cornell University, Ithaca NY 14853. Email: amitk@cs.cornell.edu. Supported in part by the ONR Young Investigator Award and NSF CAREER Award of Jon Kleinberg.

has been exploited in the design of protocols for distributed systems [5, 16]. Thus, the spread of a disease (or a computer virus, or a rumor) can be investigated by studying the *time-respecting* paths in a network.

Scheduled Transportation Networks. Finally, many of the same issues arise in the context of scheduled transportation, such as airline travel [3]. We may be given a network of airports, with edges labeled by the time(s) at which flights depart and arrive; the time-respecting paths are those that can be feasibly used by a traveler in this network.

Formally, we say that a *temporal network* is an undirected graph $G = (V, E)$ in which each edge e is annotated with a *time label* $\lambda(e)$ specifying the time at which its two endpoints "communicated." Thus, one can view a temporal network as the pair (G, λ), where λ is a function from the edge set to the real numbers; we refer to λ as a *time labeling* of G. A path P in G is called *time-respecting* if the labels on its edges are non-decreasing. Our definition is simple but robust; by the use of direct "gadget" reductions, we can model many other natural types of temporal labelings. For example, in Section 2, we show how to encode a model in which the graph is directed and each edge has separate "departure" and "arrival" times.

The central motivation for our work is the following question: how do the basic combinatorial and algorithmic properties of graphs change when we impose this additional temporal condition? The notion of a *path* is intrinsic to many of the most fundamental algorithmic problems on graphs; spanning trees, connectivity, flows, and cuts are some examples. When we focus on time-respecting paths in place of arbitrary paths, many of these problems acquire a character that is different from the traditional setting, but very rich in its own right. In particular, we provide results on two types of problems for temporal networks: *connectivity problems*, in which we seek disjoint time-respecting paths between pairs of nodes; and *inference problems*, in which we seek to reconstruct a partially specified time labeling of a network in a manner consistent with an observed history of information flow. We describe these in detail below.

Background. There is a large literature on gossiping and broadcasting algorithms in networks; see [9] for a survey. Two paradigmatic problems in this area are (i) the *Telephone Problem* [2, 4, 8, 18], in which we seek a way for n individuals to each transmit a distinct piece of information to everyone else using the minimum number of person-to-person phone calls; and (ii) the *Minimum Broadcast Time* (see [15] and the references therein), in which we seek a way for a designated source node in a graph to transmit a piece of information to all other nodes in the minimum number of parallel rounds of node-to-node communication. Note the fundamental difference between this body of work and the types of problems we will be considering. In designing a gossiping or broadcasting algorithm, one seeks to *schedule* the times at which information crosses edges of a network so as to optimize a particular objective function. In the present work, we are *given* the times at which communication has occurred, and study properties of the full history of this communication. Thus, our analysis has a more "diagnostic" character.

A network model essentially equivalent to the one we study here was proposed recently by Berman [3], in what appears to be the first work on the subject. He considered a model that he termed *scheduled networks*: directed networks in which each edge has separate departure and arrival times. As mentioned above, this can be encoded within our model of temporal networks. He gave an algorithm to compute all nodes and edges reachable from a given root by time-respecting paths, and he showed that the max-flow min-cut theorem holds, with unit capacities, for time-respecting paths. Both of these problems can be reduced to equivalent problems on standard directed graphs (with no time labels), and in particular, the max-flow min-cut result follows from this reduction; the concern in [3] was with finding algorithms that were more efficient than the reduction would allow.

The Present Work: Connectivity Problems. A common question in the analysis of gossip protocols is the following. Suppose that a node t has learned a piece of information originally possessed by a node s. If we are concerned that some nodes may be faulty and corrupt information, we can ask: for some value of k, are there k (internally) node-disjoint time-respecting paths from s to t? If this is the case, we can be more confident in the accuracy of the information t has learned, since it came along several "independent" trajectories through the network [14, 16]. In standard graphs, the existence of node-disjoint paths is characterized by Menger's Theorem [13]: the maximum number of node-disjoint s-t paths is equal to the minimum number of nodes needed to separate s from t. But there is no analogue of this theorem for arbitrary temporal networks. As observed in [3], the maximum number of node-disjoint time-respecting paths from s to t can be strictly less than the minimum number of nodes whose deletion leaves no time-respecting s-t path. Moreover, finding the maximum number of disjoint time-respecting paths is NP-complete.

The breakdown of a Menger-like theorem can be seen quite simply on the graph $\mathcal{Z}$ depicted in Figure 1, with the indicated time labeling. There are no two disjoint time-respecting paths from v_1 to v_4, but after deleting any one node (other than v_1 or v_4), there still remains a time-respecting v_1-v_4 path. Despite this counter-example, we can ask: for which graphs *does* the analogue of Menger's Theorem hold, under all time labelings? Specifically, let us say that a graph $G = (V, E)$ is *Mengerian* if for all time labelings $\lambda : E \to \mathbb{R}$ and all $s, t \in V$, the maximum number of node-disjoint time-respecting paths from s to t equals the minimum number of nodes whose deletion leaves no time-respecting s-t path. Understanding this property can provide insight into the relation between disjoint paths and node separators in temporal networks.

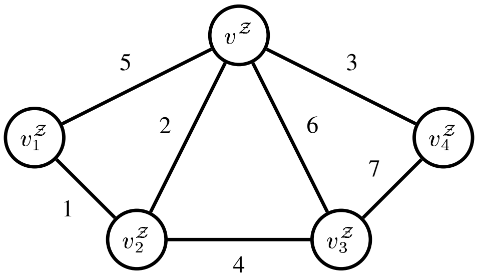

Figure 1: The graph $\mathcal{Z}$ with the labeling $\lambda_{\mathcal{Z}}$

We provide a precise characterization of Mengerian graphs: an arbitrary graph G is Mengerian if and only if it does not contain a subdivision of $\mathcal{Z}$. Thus, the Mengerian property has a characterization in the spirit of Kuratowski's Theorem expressing planarity in terms of excluded subdivisions; $\mathcal{Z}$ is in a sense the unique "obstacle" to the property. Although the statement of the theorem is quite direct, the proof arrives at the min-max relation through a non-trivial structural characterization. Moreover, as a consequence of the proof, we develop a polynomial-time algorithm to find the maximum number of disjoint time-respecting paths between a given source and sink in a time-labeled Mengerian graph.

We also consider a special case of the general disjoint paths problem — in which we seek disjoint paths connecting a constant number of terminal pairs — in the setting of temporal networks. The tractability of this problem was raised as an open question by Berman [3]. We resolve this question in two different respects. First, we show that in an (undirected) temporal network, it is NP-complete in general to decide whether there exist two disjoint time-respecting paths between a given source and sink. On the other hand, in a directed acyclic graph with a time labeling λ, there is a polynomial-time algorithm for the disjoint time-respecting paths problem for any constant number of terminal pairs. This generalizes a result of Fortune, Hopcroft, and Wyllie [6], who provided such an algorithm for standard (unlabeled) directed acyclic graphs.

We prove a number of other complexity results on connectivity problems as well; in particular, it is NP-complete to find a minimum cardinality node separator whose removal leaves no time-respecting path from a source to a sink; and it is NP-complete to approximate the maximum number of node-disjoint time-respecting paths between a given source and sink to within a factor of $|E|^{1/5-\varepsilon}$.

The Present Work: Inference Problems. A different type of problem arises when we have a temporal network with only partial information about the time labeling, and we seek to *infer* values of time labels based on additional data. For example, we may believe that communication has taken place in a network in such a way that a certain set S of nodes has learned a certain piece of information; we wish to test the feasibility of this hypothesis by reconstructing a history for the communication in which each node in S receives the information along a time-respecting path.

A general statement of this problem looks as follows. Suppose we are given a graph $G = (V, E)$, and each edge e is labeled with an interval I_e; the meaning is that we know communication took place on edge e at some time in the interval I_e, but we do not know exactly when. (Note that setting I_e to be a single point indicates precise knowledge of the time of communication.) We are also given a *root node* r, a set $P \subseteq V \setminus \{r\}$ of *positive nodes*, and a disjoint set $N \subseteq V \setminus \{r\}$ of *negative nodes*. We are told that the nodes in P learned a piece of information originating at r, and that the nodes in N did not learn this information. We are given no information about nodes in $V \setminus (P \cup N)$. The problem is:

> Does there exist a time labeling λ with $\lambda(e) \in I_e$ for each e (i.e. consistent with our partial data), there is a time-respecting path from r to each node in P, and there is no time-respecting path from r to any node in N?

We will call this the problem of *Reachability Inference*.

We provide a polynomial-time algorithm for Reachability Inference on general temporal networks. Roughly, the algorithm works by maintaining vertex labels to all vertices of G — these labels provide bounds on the earliest time at which any time-respecting path from the root may reach these vertices. Then the algorithm builds a tree rooted at r and assigns labels to the edges of the tree such that all paths in the tree originating at r are time-respecting. We show that there exists a solution to the inference problem if and only if the tree can be extended to contain all nodes of P and no node of N.

2 Preliminaries

Scheduled Networks. We begin by illustrating a reduction of Berman's *scheduled network* model [3] to our model. Suppose we have a directed graph G' in which each arc e has two time labels: a *departure time* $\lambda^-(e)$, and a larger *arrival time* $\lambda^+(e) > \lambda^-(e)$. Since the edge thus causes a delay of $\lambda^+(e) - \lambda^-(e)$, we will refer to such an edge as a *delay edge*. A path P in this model is time-respecting if, for consecutive arcs e and e' on P, we have $\lambda^+(e) \leq \lambda^-(e')$. We may view this as modeling the schedule of an airline flight along this edge, for example, or the beginning and end of a file transfer from one process to another.

We model G' with a temporal network G as follows. For each arc $e = (u, v)$ of G, we construct two undirected edges $e^- = (u, w_e)$ and $e^+ = (w_e, v)$, where w_e is a new vertex. We define $\lambda(e^-) = \lambda^-(e)$ and $\lambda(e^+) = \lambda^+(e)$. It is easy to verify that a path between two nodes in G' is time-respecting if and only if the corresponding path between the two nodes in G is time-respecting. The construction also preserves the disjointness of paths. In section 3, we will occasionally use delay edges in our constructions, meaning in fact the replacement described above.

A Reduction to Standard Directed Graphs. We also mention a construction that turns certain basic questions about temporal networks into questions about standard directed graphs. In particular, it implies a min-max theorem for *edge-disjoint* time-respecting paths [3].

Given a temporal network $G = (V, E)$, we first convert it into an equivalent "directed temporal network" G' in a standard way; by a directed temporal network, we mean simply a directed graph with a single time label on each edge, and we define the notion of a "time-respecting directed path" in the obvious fashion. To convert G to G', each undirected edge $e = (v, w) \in E$ with label $\lambda(e)$ is replaced by the following gadget: we create two new vertices x_e and y_e and add directed edges (v, x_e), (y_e, v), (x_e, y_e), (w, x_e), (y_e, w) with labels $\lambda(e)$ to E.

From our directed temporal network G', we now build an (unlabeled) directed graph G'' as follows. For each arc a of G', we create two vertices u_a and w_a, and a directed edge (u_a, w_a). Also, if a and b are arcs of G', represented by directed edges (u_a, w_a) and (u_b, w_b) in G'', we add an edge (w_a, u_b) if there is a time-respecting directed path that uses a immediately followed by b. An example of this construction is shown in Figure 2.

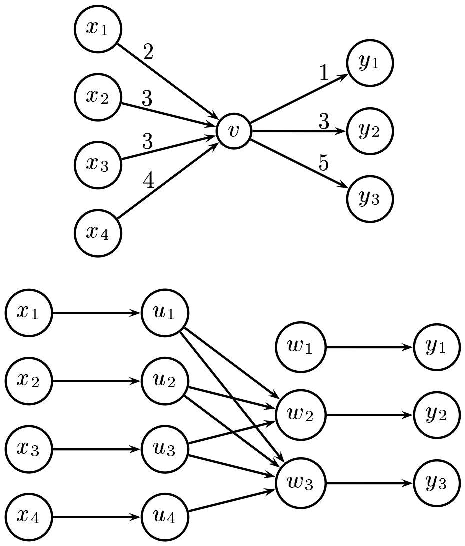

Figure 2: A vertex v and its replacement. u_i and w_j are short for u_{a_i} and w_{a_j}. The arcs are $a_i = (x_i, v)$ or $a_j = (v, y_j)$

Now, it is easy to verify that reachability in G under time-respecting paths corresponds naturally to reachability in G'', and the natural mapping of paths preserves edge-disjointness and the sizes of edge separators. Using breadth-first search in G'' we can answer questions like, "Starting at node u at time t, how early can one reach node v using a time-respecting path?" Moreover, if we add a super-source and super-sink to G'', the max-flow min-cut theorem for (unlabeled) directed graphs implies

Proposition 2.1 (Berman [3]) *For a temporal network G with nodes s and t, the maximum number of edge-disjoint time-respecting paths from s to t is equal to the minimum number of edges whose deletion leaves no time-respecting s-t path.*

However, this reduction does not imply anything about node-disjoint paths; and it is clear that there is no simple relation between temporal networks and standard directed graphs preserving the notion of node-disjointness, given that the Mengerian property for temporal networks has a non-trivial characterization. The reduction is also not useful for reasoning about inference problems, since one can only construct the directed graph associated with a temporal network G *given* the labeling of the edges of G.

3 Connectivity Problems and Mengerian Graphs

Given a temporal network with underlying graph $G = (V, E)$ and time labeling λ, and two nodes $s, t \in V$, we say that a collection of time-respecting s-t paths is *internally node-disjoint* (or simply *node-disjoint*, by slight abuse of terminology) if the paths share no nodes other than s and t. We use $p^v_{G,\lambda}(s, t)$ to denote the maximum cardinality of such a collection. A set $S \subseteq (V \setminus \{s, t\}) \cup E$ is called an *s-t separator* (or *s-t vertex-separator* or *cut-set*) with respect to λ, if it meets every time-respecting s-t path in an edge or vertex.[1] The minimum size of any s-t separator with respect to λ is denoted by $c^v_{G,\lambda}(s, t)$.

In the introduction, we discussed the temporal network in Figure 1, with underlying graph $\mathcal{Z}$ and the indicated time labeling; it has $p^v_{\mathcal{Z},\lambda}(v_1, v_4) = 1$, but $c^v_{\mathcal{Z},\lambda}(v_1, v_4) = 2$. This is in sharp contrast to the statement of Menger's Theorem, which shows that these two quantities are equal for all *unlabeled* graphs (or equivalently, for all graphs in which $\lambda(e) = 0$ for all e). We can in fact generalize this example to provide a family of temporal networks $\{\mathcal{Z}_k : k = 2, 3, \ldots\}$ with nodes s_k, t_k for which $p^v_{\mathcal{Z}_k,\lambda}(s_k, t_k) = 1$, but $c^v_{\mathcal{Z}_k,\lambda}(s_k, t_k) = k$. The graph $\mathcal{Z}_k$ has $\Theta(k^3)$ nodes, indicating a gap of $\Omega(|V|^{1/3})$ between the maximum number of disjoint paths and the minimum size of a vertex-separator in the worst case.

For the general idea underlying this construction, reconsider the three vertices $v^{\mathcal{Z}}, v_2^{\mathcal{Z}}, v_3^{\mathcal{Z}}$ in Figure 1, which from now on — in a slight abuse of terminology — shall be called the *inner vertices*. Every time-respecting $v_1^{\mathcal{Z}}$-$v_4^{\mathcal{Z}}$ path has to visit at least two inner vertices, ensuring that there can be no two disjoint paths. On the other hand, there is a time-respecting $v_1^{\mathcal{Z}}$-$v_4^{\mathcal{Z}}$ path through any set of 2 inner vertices, so that any $v_1^{\mathcal{Z}}$-$v_4^{\mathcal{Z}}$ separator must have size at least 2.

We generalize this idea to a graph with $2k - 1$ inner vertices, and label the edges so that every time-respecting s-t path visits at least k of these vertices (ensuring again that there are no two disjoint s-t paths). On the other hand, we make sure that there is a time-respecting s-t path through any

[1]The inclusion of edges is only necessary in the special case that there is an edge (s, t) in G. In all other cases, we could simply include one of the endpoints of an edge e instead of e itself.

set of k inner vertices, so every s-t separator must have size at least k.

The graph with $2k-1$ inner vertices will be denoted by $\mathcal{Z}_k$, and produce a gap of k. The inner vertices are labeled $1,\ldots,2k-1$. Vertices $1,\ldots,k$ are connected to s with edges labeled $\lambda_k((s,i)) = 2i-1$, vertices $k,\ldots,2k-1$ to t with edges labeled $\lambda_k((i,t)) = 2i-k$. Between any pair of inner vertices $v < w$, we add k delay edges with labels $(v, 2w-v-1), (v+1, 2w-v), \ldots, (v+k-1, 2w-v+k-2)$. As delay edges are implemented by auxiliary vertices, the resulting graph is still simple. The important properties of the resulting graph Z_k and labeling λ_k are summarized in the following lemma:

Lemma 3.1 $c^v_{\mathcal{Z}_k,\lambda_k}(s,t) = k$, *and* $p^v_{\mathcal{Z}_k,\lambda_k}(s,t) = 1$.

Proof Sketch. To prove that $c^v_{\mathcal{Z}_k,\lambda_k}(s,t) \geq k$, one considers any separator of size smaller than k, and verifies that the path visiting all other vertices in ascending order and always taking the earliest possible delay edge is indeed time-respecting.

To prove that $p^v_{\mathcal{Z}_k,\lambda_k}(s,t) \leq 1$, we prove that every time-respecting s-t path visits at least k inner vertices. The key property (established by induction on v) is that a time-respecting s-v path reaching the inner vertex v at time τ must visit at least $2v-\tau$ inner vertices including v. ■

Corollary 3.2 *The graphs $\mathcal{Z}_k$ have an $\Theta(\sqrt[3]{n})$ gap between the maximum number of disjoint time-respecting s-t paths and the minimum size of an s-t separator.*

In the notation above, we can say that a graph G is *Mengerian* if for all time labelings $\lambda : E \to \mathbb{R}$ and all $s,t \in V$, we have $p^v_{G,\lambda}(s,t) = c^v_{G,\lambda}(s,t)$. Our next goal in this section is to prove the following characterization theorem. We first recall two definitions from graph theory. We say that a graph H' is a *subdivision* of a graph H if H' can be obtained from H by replacing each edge with a chain of degree-2 vertices. We say that G contains H as a *topological minor* if it contains a subgraph isomorphic to a subdivision of H.

Theorem 3.3 *$G = (V,E)$ is Mengerian if and only if it does not contain $\mathcal{Z}$ as a topological minor.*

The "only if" direction is easier:

Lemma 3.4 *If G contains $\mathcal{Z}$ as a topological minor then G is not Mengerian.*

Proof Sketch. Given a graph G containing $\mathcal{Z}$ as a topological minor, we can use the subdivision of $\mathcal{Z}$ to label the edges of G, and choose s and t, so that $p^v_{G,\lambda}(s,t) < c^v_{G,\lambda}(s,t)$, Specifically, fix a subgraph H of G isomorphic to a subdivision of $\mathcal{Z}$. We define s to be the node representing $v_1^{\mathcal{Z}}$ in H, and t to be the node representing $v_4^{\mathcal{Z}}$ in H. Each edge of G lying on a path in H that corresponds to an edge e of $\mathcal{Z}$ will receive the label that e is assigned in Figure1. All other edges entering t receive the label 0, and all other edges that do not enter t receive the label 8. (If there is an edge joining s and t in G, it can be labeled arbitrarily). It is then easy to verify that $p^v_{G,\lambda}(s,t) < c^v_{G,\lambda}(s,t)$. ■

The harder statement here is the "if" direction, which we prove by induction on the number of edges of G. We begin with some lemmas that facilitate the proof. First, if the maximum degree of G is 3, then a set of paths is internally node-disjoint if and only if it is edge-disjoint; applying Proposition 2.1, we have

Lemma 3.5 *Let G be a graph such that the degree of none of its vertices exceeds* 3. *Then G is Mengerian.*

Now, if G contains a cut-node v (a vertex whose deletion disconnects it), then one can show that G is Mengerian if and only if each of its two-connected components is. This provides an easy way to apply induction when G is not two-connected.

Lemma 3.6 *Let G be a non-Mengerian graph that is not two-connected. Then G contains a proper subgraph that is non-Mengerian.*

Proof. Suppose G is non-Mengerian, and consider a labeling λ and a choice of s and t that violate the Mengerian property. Then s and t must lie in the same two-connected component C of G; for otherwise, there are no two vertex-disjoint time-respecting paths joining them, and we cannot have $p^v_{G,\lambda}(s,t) < c^v_{G,\lambda}(s,t)$. But now one can verify that the subgraph C must also be non-Mengerian, as no simple s-t path can visit nodes outside of C. ■

We now develop a structural property of graphs that do not contain a subdivision of $\mathcal{Z}$. Given a two-connected graph $G = (V,E)$, two vertices $v,w \in V$, and a natural number $d \geq 1$, we say that G is (v,w,d)-separable if

(i) v and w each have degree d.

(ii) Either $G \setminus \{v,w\}$ consists of d connected components; or $G \setminus \{v,w\}$ consists of $d-1$ connected components, and (v,w) is an edge of G.

Figure 3 depicts a (v,w,d)-separable graph; we will refer to the subgraphs induced on the connected components of $G\setminus\{v,w\}$ as the *lobes* of G, and denote them by $G_1,\ldots,G_d$. We use e_i (resp. f_i) to denote the edge from v (resp. w) into G_i; note that one of the G_i may be empty, in the case that $(v,w) \in E$, and then we have $e_i = f_i$.

Lemma 3.7 *Let G be a two-connected graph with at least one vertex of degree at least* 4. *Then G contains $\mathcal{Z}$ as a topological minor, or there are $v,w \in V$ and a number $d \geq 4$ so that G is (v,w,d)-separable.*

Proof. Let v be a vertex of degree $d \geq 4$, and $v_1, \ldots v_d$ be its neighbors. Consider the graph $G \setminus \{v\}$, the result of deleting v and all its incident edges. As G was two-connected,

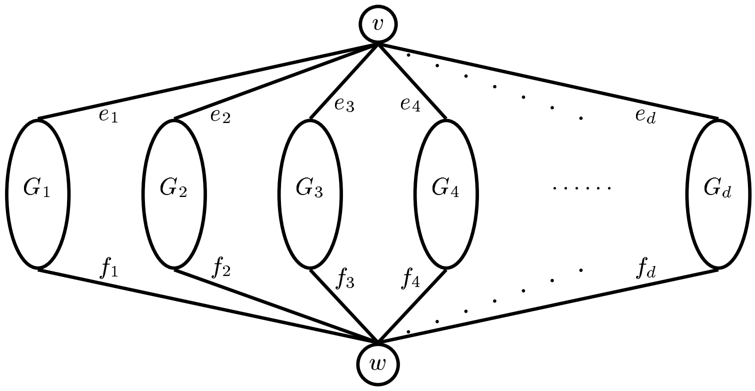

Figure 3: The form of a (v, w, d)-separable graph

$v_1, \ldots, v_d$ are still connected in $G \setminus \{v\}$. Let T be any inclusion-wise minimal Steiner tree spanning $v_1, \ldots, v_d$ in $G \setminus \{v\}$.

If no vertex in T has degree exceeding 2, then T is a path. Let v_i, v_j, v_k, v_l be the first four vertices from $v_1, \ldots, v_d$ appearing on T. Then, G contains $\mathcal{Z}$ as a topological minor, with v_i, v_j, v_k, v_l as the images of $v_1^{\mathcal{Z}}, v_2^{\mathcal{Z}}, v_3^{\mathcal{Z}}, v_4^{\mathcal{Z}}$.

Otherwise, let w be any vertex of degree $d_w \geq 3$ in T. Each of the d_w subtrees rooted at w must contain at least one of $\{v_1, \ldots, v_d\} \setminus \{w\}$ (w might be identical to one of the v_i), because otherwise, that subtree could be deleted and T would not be minimal.

If any subtree contains two vertices $v_i, v_j, i \neq j$, let $v_k, v_l, k \neq l \neq i, j$ be vertices in different subtrees (there must be at least three subtrees). Then G contains $\mathcal{Z}$ as a topological minor. For either one of v_i, v_j lies on the path from w to the other — say, v_i lies on w–v_j — in which case we obtain $\mathcal{Z}$ by mapping $(v^{\mathcal{Z}}, v_1^{\mathcal{Z}}, v_2^{\mathcal{Z}}, v_3^{\mathcal{Z}}, v_4^{\mathcal{Z}})$ to (v, v_k, w, v_i, v_j) (and the edge $(v^{\mathcal{Z}}, v_2^{\mathcal{Z}})$ to the path v–v_l–w). Or there must be a last vertex w' that is common to the paths w–v_i and w–v_j, in which case we map $(v^{\mathcal{Z}}, v_1^{\mathcal{Z}}, v_2^{\mathcal{Z}}, v_3^{\mathcal{Z}}, v_4^{\mathcal{Z}})$ to (v, v_k, w, w', v_i) (and the edges $(v^{\mathcal{Z}}, v_2^{\mathcal{Z}})$ and $(v^{\mathcal{Z}}, v_3^{\mathcal{Z}})$ to the paths v–v_l–w and v–v_j–w', respectively).

From now on, we may assume that each subtree rooted at w contains exactly one vertex v_i, so the subtrees are in fact paths, i.e. T consists of disjoint paths $P_1, \ldots, P_d$ from w to $v_1, \ldots, v_d$, respectively. In the case that w coincides with one of the v_i, the path P_i is empty.

Consider the graph $G' = G \setminus \{v, w\}$, obtained by removing from G the vertices v and w and all their incident edges. Define the graph G_i to be the connected component of G' containing v_i (if $w = v_i$ for some i, then the corresponding G_i is empty).

The set of all G_i must cover G'. For assume that there were another component C of G' that contained none of the v_i. No vertex of C can be adjacent to v (because then, this vertex would be one of the v_i), and by construction, there is no edge between C and any G_i. So, all paths between C and v must go through w, and G would not have been two-connected.

If $G_i = G_j$ for some $i \neq j$, there must be a path P between v_i and v_j. Let w_i be the last vertex of P that also lies on P_i, and w_k the first vertex appearing after w_i on P that lies on some path P_k for $k \neq i$ (k and j might be identical, but don't have to be). With $l \neq i, k$ (and $v_l \neq w$), we get $\mathcal{Z}$ as a topological minor by mapping $(v^{\mathcal{Z}}, v_1^{\mathcal{Z}}, v_2^{\mathcal{Z}}, v_3^{\mathcal{Z}}, v_4^{\mathcal{Z}})$ to (v, w_i, w_k, w, v_l) (and the edge $(v^{\mathcal{Z}}, v_3^{\mathcal{Z}})$ to the path v–v_m–w, for some $m \neq i, k, l$).

The only step left to establish the lemma is to show that the existence of several edges (w, x) and (w, y) with $x, y \in G_i$ would again yield $\mathcal{Z}$ as a topological minor in G. Assume that two such edges exist. Then, $x \neq y$, because G is simple. Consider paths P_x and P_y between x and v_i (y and v_i, respectively), and let z be the first vertex of P_x that lies on P_y (such a vertex must exist, because both paths meet v_i).

Because $x \neq y$, the vertex z cannot be equal to both — assume without loss of generality that $z \neq y$. Let $j, k \neq i$. We then obtain $\mathcal{Z}$ as a topological minor, mapping nodes $(v^{\mathcal{Z}}, v_1^{\mathcal{Z}}, v_2^{\mathcal{Z}}, v_3^{\mathcal{Z}}, v_4^{\mathcal{Z}})$ to (w, y, z, v, v_j) (and the edges $(v^{\mathcal{Z}}, v_2^{\mathcal{Z}})$ and $(v^{\mathcal{Z}}, v_3^{\mathcal{Z}})$ to the paths w–x–z and w–v_k–v, respectively).

If none of the above cases yielding $\mathcal{Z}$ applies, then G is (v, w, d)-separable, completing the proof. ■

Using the structural property of Lemma 3.7, we can now complete the proof of Theorem 3.3.

Proof of Theorem 3.3. One direction has already been established as Lemma 3.4. For the converse direction, assume that there exist non-Mengerian graphs that do not contain $\mathcal{Z}$ as a topological minor, and let $G = (V, E)$ have the smallest number of edges among all such graphs (G need not be unique).

Then, G must be two-connected according to Lemma 3.6, and it must contain at least one vertex v of degree greater or equal to 4 according to Lemma 3.5. We can therefore apply Lemma 3.7, and because G does not contain $\mathcal{Z}$ as a topological minor by assumption, it must be (v, w, d)-separable for vertices $v, w \in V$ and some $d \geq 4$.

Let λ, s and t be such that $c^v_{G,\lambda}(s, t) > p^v_{G,\lambda}(s, t)$, and $\mathcal{P}$ a maximum set of vertex-disjoint time-respecting s-t paths.

We distinguish between four cases based on the locations of s and t.

(1) If s and t both lie in the same lobe G_i, consider the graph G', defined as the subgraph induced by $G_i \cup \{v, w\}$. If there is a time-respecting v-w path in $G \setminus G_i$ starting at v at time $\lambda(e_i)$ and arriving at w at time no later than $\lambda(f_i)$, or starting at w at time $\lambda(f_i)$ and arriving at v no later than $\lambda(e_i)$, we add an edge (v, w) labeled $\lambda'((v, w)) = \min(\lambda(e_i), \lambda(f_i))$ (if the edge (v, w) existed in G, we just relabel it). If no such path exists, and the edge (v, w) existed in G, we remove it from G'. All other labelings stay the same. We prove that $c^v_{G',\lambda'}(s, t) = c^v_{G,\lambda}(s, t)$ and $p^v_{G',\lambda'}(s, t) = p^v_{G,\lambda}(s, t)$ (this obviously yields a contradiction, G' being smaller than G).

For the first equation, let S be a set of vertices separating s from t in G'; we show that S also separates s from t in G. Let P be any time-respecting s-t path in G, with vertices $V(P)$. If $V(P) \subseteq V(G')$, then P is a path in G_i and must meet S. Otherwise, P must pass through v and w, and we obtain another time-respecting s-t path P' by replacing the subpath vPw with the edge (v, w).[2] Because $P' \subseteq G'$, it meets S, and because $V(P') \subseteq V(P)$, P must also meet S.

For the second equation, note that $\mathcal{P}$ can contain at most one path P not entirely in G_i. Applying the above replacement of vPw by (v, w) to P (if it exists) yields a set $\mathcal{P}'$ of time-respecting node-disjoint s-t paths in G' with size $|\mathcal{P}'| = |\mathcal{P}|$. Conversely, if $\mathcal{P}'$ is a set of time-respecting node-disjoint s-t paths in G', we can replace the edge (v, w) (if it is part of one of the paths) by the time-respecting v-w path that we assumed to exist in G when we added the edge (v, w), obtaining a set of paths of the same size.

(2) If s lies in some lobe G_i, and t in another lobe G_j, each time-respecting s-t path must pass through v or w; that is, there can be at most two disjoint paths, and deleting v and w suffices to disconnect s from t. For G to be non-Mengerian, there would have to be exactly one disjoint time-respecting s-t path, but each s-t separator would have to have size 2. In particular, neither v nor w separate s from t, so there must be time-respecting s-t paths P_v, P_w that use v (resp. w), but not w (resp. v).

Consider the graph G' induced by $G_i \cup G_j \cup \{v, w\}$, without the edge (v, w), if it existed in G. Because G was minimal by assumption, G' must have an s-t separator of size 1, i.e. some vertex $\hat{v}$.

$\hat{v}$ must meet both P_v and P_w, because both of these paths lie entirely in G'. Therefore $\hat{v}$ must lie in G_i or G_j — by symmetry, we will assume that $\hat{v} \in G_i$. Now, let P be a time-respecting s-t path in G that does not meet $\hat{v}$. Then, P must pass through both v and w. Assume without loss of generality that v appears on P before w. Let P' be the path obtained by replacing vPt with vP_vt. P' is a time-respecting s-t path entirely in G', and not meeting $\hat{v}$, contradicting the fact that $\hat{v}$ is an s-t separator for G'.

[2]For a path P and two nodes $x, y \in P$, we use xPy to denote the subpath of P with ends equal to x and y.

(3) If $s = v$, and t lies in some lobe G_i (or one of the three symmetric cases $s = w$, $t = v$, or $t = w$), we can apply an argument much like the above. All s-t paths must use either e_i or f_i, so if G were non-Mengerian, we would have $c^v_{G,\lambda}(s, t) = 2$ and $p^v_{G,\lambda}(s, t) = 1$. Again, there must be paths P_e and P_f using exactly one of e_i and f_i each.

Let G' be the subgraph induced by $G_i \cup \{v, w\}$. If there is a time-repecting v-w path in $G \setminus G_i$ reaching w at time $\lambda(f_i)$ or earlier, we add an additional edge (v, w) labeled $\lambda'((v, w)) = \lambda(f_i)$. Otherwise, we remove the edge (v, w), if it existed in G.

Again, G' is smaller than G, and the rest of the argument is the same as in the previous case.

(4) The last remaining case is that $s = v$ and $t = w$ (or vice versa). In that case, any time-respecting s-t path passes through exactly one of the G_i. Let P_i denote such a path through G_i, if it exists. Because all paths through G_i have to use e_i, there cannot be two disjoint paths through any one G_i; and for different i, the P_i are disjoint. Because each P_i can be blocked by deleting e_i, we obtain that

$$\begin{aligned} p^v_{G,\lambda}(s, t) &= \left| \left\{ i \mid \begin{array}{c} \text{there is a time-respecting} \\ s\text{-}t \text{ path through } G_i \end{array} \right\} \right| \\ &\geq c^v_{G,\lambda}(s, t), \end{aligned}$$

completing the proof. ■

A polynomial-time algorithm

The proofs of Lemmas 3.5 – 3.7 and Theorem 3.3 are implicitly algorithmic; taken together, they can be used to obtain a polynomial-time algorithm for the following problem: Given a Mengerian graph G, with vertices $s, t \in V$, find a maximum set of disjoint time-respecting paths from s to t.

The proof of Lemma 3.7 provides an efficient algorithm to identify either a subdivision of $\mathcal{Z}$ in G, or v, w, and d such that G is (v, w, d)-separable. In the latter case, if $\{s, t\} = \{v, w\}$, then we can find a maximum set of disjoint paths directly. Otherwise we can identify the locations of s and t relative to the set of lobes G_i; by strict analogy with the inductive structure of Theorem 3.3, we can then "reduce" certain parts of the graph G and proceed recursively.

We thus obtain the following theorem; details of the proof are given in the full version of the paper.

Theorem 3.8 *There is a polynomial-time algorithm that takes a graph $G = (V, E)$, and vertices $s, t \in V$, and returns either a maximum set of disjoint time-respecting paths from s to t or a subgraph of G isomorphic to a subdivision of $\mathcal{Z}$.*

We also note that this approach allows us to test whether a graph is Mengerian in polynomial time, without invoking the minor-testing algorithm of Robertson and Seymour [17].

Disjoint Paths with a Fixed Number of Terminal Pairs

In the general disjoint paths problem, we are given an (unlabeled) graph $G = (V, E)$, as well as k pairs of *terminals*

$(s_1, t_1), \ldots, (s_k, t_k)$, where $s_i, t_i \in V$. We seek k node-disjoint paths $P_1, \ldots, P_k$ so that P_i joins s_i to t_i. This problem is NP-complete when k is part of the input [10], or when G is directed and $k = 2$ [6]. When k is a fixed constant, the problem is polynomial-time solvable when G is undirected, by an algorithm of Robertson and Seymour [17], and when G is a directed acyclic graph, by an algorithm of Fortune, Hopcroft, and Wyllie [6].

In [3], Berman asked about the natural extension of these results to the case of time-respecting disjoint paths: can we find k node-disjoint paths in polynomial time when k is a fixed constant? We first show that the result for undirected graphs does not carry over (assuming $P \neq NP$), even when $k = 2$, all s_i are the same, and all t_i are the same.

Theorem 3.9 *(a) Given a temporal network (G, λ), with vertices s and t, it is NP-complete to decide whether there exist two node-disjoint time-respecting s-t paths.*
(b) It is also NP-hard to approximate the maximum number of disjoint time-respecting s-t paths within a factor of $O(|E|^{1/5-\varepsilon})$ for any $\varepsilon > 0$.

Proof. The theorem is proved by a direct reduction from the hardness (resp. approximation hardness) of the bounded length vertex-disjoint paths (BLVDP) problem, which is defined as follows:

Given a directed graph $G = (V, A)$ with edge lengths $l : A \to \mathbb{N} \setminus \{0\}$, a length bound b, and two vertices $s, t \in V$, what is the maximum size of a set of vertex-disjoint s-t paths, each of whose lengths is bounded by b?

The NP-hardness of deciding whether there are two vertex-disjoint paths with lengths bounded by b was proved in [12]. [7] proved that it is NP-hard to approximate within a factor of $m^{1/2-\varepsilon}$ the maximum number of *edge-disjoint* s-t paths whose lengths are bounded by some $b = O(m^{1/2})$. Their proof also works for vertex-disjoint paths. However, their construction only produces *non-negative* edge lengths. Some small changes ensure that the edge lengths are positive, but require us to use a length bound $b = O(m^{3/2})$. We refer the reader to [7] for more details.

For the reduction to the vertex-disjoint time-respecting paths problem, we will make use of the delay edges introduced above. We construct an undirected graph $G' = (V', E')$ with labeling λ, whose vertex set will consist of the set V, and the auxiliary vertices introduced through delay edges.

For every arc $a = (v, w) \in A$ with length $l(a) > 0$, we add $b - l(a) + 1$ delay edges between v and w, labeled $(0, l(a)), \ldots, (b - l(a), b)$. Note that because $l(a) > 0$, these delay edges can only be traversed in one direction, and hence, we can think of them as directed, even though our graph is actually undirected. Also, because delay edges implicitly introduce additional vertices, the resulting graph is simple.

It is then straightforward to verify that there exist k vertex-disjoint time-respecting paths between s and t in G' if and only if there exist k vertex-disjoint paths in G whose length is bounded by b, proving part (a) of the theorem.

The number of edges in G' is $|E'| \leq b|E| = O(m^{5/2})$, and we therefore get a gap of $m^{1/5-\varepsilon}$ for part (b), completing the proof. ■

However, by generalizing the "pebbling game" of Fortune, Hopcroft, and Wyllie [6] to the setting of time-respecting paths, we obtain the following result. Again, the proof is given in the full version.

Theorem 3.10 *There is a polynomial-time algorithm for the time-respecting disjoint paths problem with a fixed number of terminal pairs in a directed acyclic graph.*

4 Inference Problems

Thus far, we have always been given a graph and a complete time labeling, and we have sought to decide certain properties of the labeled graph. We now turn to the class of *inference problems*, in which we are given an incomplete time labeling; the task is then to extend the labeling to ensure that the graph has a certain property, or conclude that such an extension is not possible.

In this section, we provide a polynomial-time algorithm for the Reachability Inference Problem discussed in the introduction. Consider an undirected graph $G = (V, E)$ rooted at r. Let $n = |V|$ and $m = |E|$. Each edge e of the graph is assigned an interval I_e. This interval can be closed, open or semi-open — however, for the purpose of the following exposition, we will assume without loss of generality that all intervals I_e are closed[3]. We are also given two disjoint sets $P, N \subseteq V \setminus \{r\}$. We wish to find a time labeling λ so that there is a time-respecting path from r to each node of P, and to no node of N. We will call such a labeling a *valid* labeling.

The algorithm will assign vertex labels $\mu(v)$ to each vertex of V — this will denote the fact that the labeling λ should be such that there is no time-respecting r-v path which arrives at v at time $\mu(v)$ or earlier. For each edge e, let $lower(e)$ and $upper(e)$ be the lower and upper limits of the interval I_e assigned to e. The algorithm is shown in Figure 4.

Let us first analyze the running time of this algorithm. Call an edge *sufficient* if it is not insufficient. Note that once an insufficient edge becomes sufficient, it remains sufficient (because μ values can only increase). Therefore, step 2 can be implemented to take $O(m \log n)$ time. Step 5 can be implemented in $O(m + n \log n)$ time, so the running time of the algorithm is $O(m \log n)$.

Let us now prove the correctness of the algorithm. We maintain the invariant that any valid labeling of E cannot contain a time-respecting r-v path which reaches v on or before $\mu(v)$ (i.e., the label of the last edge on this path must be greater than $\mu(v)$). This is clearly true after step 1. Consider

[3] In fact, edges can be labeled with any set, so long as the set has an appropriate representation, and membership can be decided within the desired complexity bounds.

```
Algorithm 1 :
   1.  Initialize μ(v) = ∞ for all v ∈ N and μ(v) = −∞ for all v ∈ V \ N.
       Call an edge e = (u,v) ∈ E insufficient if μ(u) < lower(e) and upper(e) ≤ μ(v).
   2.  While there is an insufficient edge e = (u,v) in E do
          μ(u) ← lower(e).
       If μ(r) = ∞, there is no valid labeling of E.
   3.  Initialize T to be the root r (T will remain a tree rooted at r).
       Let ε be a sufficiently small positive constant (specified below).
   4.  If v is a vertex in T, define t_v to be the label λ(e) of
          the last edge e on the unique r-v path in T, and set t_r = −∞.
       Call an edge e = (u,v) admissible if t_u ≤ upper(e) and μ(v) < upper(e).
       Let δ(T) denote the set of edges e ∈ E such that
          T contains exactly one end-point of e.
   5.  While δ(T) contains an admissible edge do
          Let e = (u,v), u ∈ T, be the admissible edge in δ(T) which attains the
                minimum value of the quantity q(e) = max(μ(v) + ε, t_u, lower(e)).
          Add e to T and set λ(e) ← q(e).
   6.  If P ⊆ T then
          For all unlabeled edges e, set λ(e) ← lower(e).
          λ is a valid labeling of E.
       Else
          There is no valid labeling of E.
```

Figure 4: The algorithm for label assignments

an insufficient edge $e = (u, v)$. If there is a labeling such that a time-respecting r-u path reaches u on or before $lower(e)$, then any label on the edge e will result in a time-respecting r-v path which reaches v on or before $\mu(v)$, a contradiction. Thus, we can make $\mu(u)$ at least $lower(e)$. So, the invariant remains true after step 2. Also, all edges are sufficient after step 2.

For a sufficiently small constant ϵ, we can consider without loss of generality valid labelings λ' that satisfy the following property: any time-respecting r-v path with labeling λ' does not reach a vertex v before $\mu(v) + \epsilon$.

In steps 3–5, we maintain the following invariant. Let λ' be a valid labeling of E. Then, for any vertex $v \in T \setminus \{r\}$, the earliest time at which a time-respecting (with respect to λ') r-v path can arrive at v is at least t_v. Clearly, this holds after step 3.

We now show that the invariant holds after each iteration of the while-loop in step 5. Let $e = (u, v) \in \delta(T)$ be an admissible edge which attains the minimum, and suppose that T satisfies the invariant before e is added to it. First note that the unique path from r to v in $T \cup \{e\}$ is time-respecting, satisfies the condition that it does not reach v before $\mu(v) + \epsilon$, and has $q(e) \in I_e$ (because e is admissible). We need to show that $T \cup \{e\}$ also satisfies the invariant. Suppose not. Then, there exists a valid labeling λ' and a time-respecting r-v path P_v with respect to λ' which arrives at v before $t_v = q(e)$. This path must contain an edge $e' = (u', v') \in \delta(T)$, $u' \in T$. We claim that e' is admissible. By the invariant, P_v does not reach u' before $t_{u'}$, and because P_v is time-respecting, $t_{u'} \leq upper(e')$. Further, the fact that λ' is valid implies that P_v cannot reach v' on or before $\mu(v')$, so $\mu(v') < upper(e')$. Therefore, e' is admissible.

As the algorithm selected e, we know that $q(e') \geq q(e)$. If $\mu(v') + \epsilon \geq q(e)$, then clearly, P_v cannot reach v' before $q(e)$ and hence cannot reach v before $q(e)$ (recall that $t_v = q(e)$). If $t_{u'} \geq q(e)$ or $lower(e') \geq q(e)$, then P_v, respecting time, cannot reach v before $q(e)$ (because it reaches v' no earlier than $q(e)$). Thus, we get a contradiction and the invariant holds after step 5.

Claim 4.1 *Let λ' be any valid labeling of E. Then, there exists a time-respecting r-v path with respect to λ' only if $v \in T$ (here T is the tree obtained after step 5).*

Proof. Suppose the claim is false, and let λ' be a valid labeling such that there exists a time-respecting r-v path P_v with $v \notin T$. Then, there is an edge $e' \in \delta(T) \cap P_v$. As we argued above, e is an admissible edge and step 5 should not have terminated. Thus, we get a contradiction. ■

So, if P is not a subset of T, then there is no valid labeling. If $P \subseteq T$, then λ can be extended to all edges as shown in step 6. Indeed, none of the nodes of N are in T because these nodes have labels ∞. Further, if $(u, v) \in E$ such that $u \in T, v \notin T$, then the fact that e is not admissible implies that $t_u > upper(e) \geq lower(e)$ or $\mu(v) \geq upper(e)$. In the latter case, the fact that e is sufficient implies that $\mu(u) \geq lower(e)$ and so, $t_u > lower(e)$. Thus, the set of nodes reachable from r is exactly equal to those in T, and the algorithm is correct.

5 Final Remarks

A consequence of the analysis in the preceding section is the following: If (G, λ) is a temporal network with a root r, such that every node v is reachable from r by a time-respecting path, then in fact there is a directed arborescence T rooted at r such that the unique r-v path in T is time-respecting for every node v. (We will call such a structure a *time-respecting arborescence*.) This follows by applying the inference algorithm to the "trivial" case in which all nodes belong to P, and each interval I_e is a single point.

This result begins to look slightly more unexpected when one considers the failure of natural related reachability properties in temporal networks. For example, suppose we have a temporal network with graph $G = (V, E)$ and labeling λ, with the property that for every $u, v \in V$, there is a time-respecting path from u to v. Let us call such a (G, λ) a *temporally connected network*. Given a temporally connected network $G = (V, E)$ on n nodes, is there a set $E' \subseteq E$ consisting of $O(n)$ edges so that the temporal network on the subgraph (V, E') is also temporally connected? In other words, do all temporal networks have sparse subgraphs preserving this basic connectivity property? The answer to the analogous question is affirmative for standard graphs, simply by taking a spanning tree in the undirected case, or the union of two arborescences in the directed case. We can show, however, that there exist temporally connected networks for which no subgraph of fewer than $\Omega(n \log n)$ edges is temporally connected. We ask as an open question: what is the tightest function $f(n)$ for which every temporally connected network on n nodes has a subgraph on $f(n)$ edges that is also temporally connected?

We can also show that the natural analogue of Edmonds' Theorem on edge-disjoint arborescences breaks down for temporal networks, even in the following approximate sense:

Theorem 5.1 *For every constant k, there exists a temporal network with root r, so that r has k edge-disjoint time-respecting paths to each other node, but there are no two edge-disjoint time-respecting arborescences rooted at r.*

Proof. Consider the graph G_k consisting of a root r, connected to vertices $v_1, \ldots, v_{2k-1}$ by arcs labeled 0. There is an arc labeled 1 from each v_i to each v_j. In addition, we have a vertex w_S for each set $S \subseteq \{v_1, \ldots, v_{2k-1}\}$ with $|S| = k$, which has incoming arcs labeled 0 from all v_i with $i \in S$.

It is then easy to verify that there are k disjoint time-respecting paths from the root to every other vertex. However, if there were two disjoint time-respecting arborescences, one of them (call it A) could contain at most $k - 1$ of the arcs (r, v_i). For a subset S (of size k) of the indices i with $(r, v_i) \notin A$, there will be no time-respecting path from r to w_S in A. ■

As another open question, we ask: what is the tightest function $g(n)$ so that, in every temporal network on n nodes with c disjoint time-respecting paths from a root r to each other vertex, there are at least $c/g(n)$ disjoint time-respecting arborescences rooted at r?

Acknowledgments

We thank Yaron Minsky and Robbert van Renesse for valuable discussions on gossip protocols that motivated our investigation of this model.

References

[1] N. Bailey, *The Mathematical Theory of Infectious Diseases and its Applications*, Hafner Press, 1975.

[2] B. Baker, R. Shostak, "Gossips and telephones," *Discrete Mathematics* 2(1972), pp. 191–193.

[3] K. Berman, "Vulnerability of scheduled networks and a generalization of Menger's Theorem," *Networks* 28(1996), pp. 125–134.

[4] R. Bumby, "A problem with telephones," *SIAM J. Algebraic and Discrete Methods* 2(1981), pp. 18–31.

[5] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Stuygis, D. Swinehart, D. Terry, "Epidemic algorithms for replicated database maintenance," *Proc. ACM Symp. on Operating Systems Principles*, 1987.

[6] S. Fortune, J. Hopcroft, J. Wyllie, "The directed subgraph homeomorphism problem," *Theoretical Computer Science* 10(1980), pp. 111–121.

[7] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, M. Yannakakis, "Near-Optimal Hardness Results and Approximation Algorithms for Edge-Disjoint Paths and Related Problems," *Proc. ACM Symposium on Theory of Computing*, 1999, pp. 19–28.

[8] A. Hajnal, E. Miller, E. Szemerédi, "A cure for the telephone disease," *Canadian Math. Bulletin* 15(1972), pp. 447–450.

[9] S. Hedetniemi, S. Hedetniemi, A. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks* 18(1988), pp. 319–349.

[10] R.M. Karp, "On the computational complexity of combinatorial problems," *Networks* 5(1975), pp. 45–68.

[11] K. Kuratowski, "Sur le problème des courbes gauches en topologie," *Fundam. Math.* 15 (1930), pp. 271–283.

[12] C. Li, T. McCormick, D. Simchi–Levi, "The complexity of finding two disjoint paths with min-max objective function," *Discrete Appl. Math.* 26(1990), pp. 105–115

[13] K. Menger, "Zur allgemeinen Kurventheorie," *Fundam. Math.* 19(1927), pp. 96–115.

[14] Y. Minsky, personal communication, April 1999.

[15] R. Ravi, "Rapid rumor ramification: Approximating the minimum broadcast time," *Proc. IEEE FOCS*, 1994.

[16] R. van Renesse, Y. Minsky, M. Hayden, "A gossip-style failure-detection service," *Proc. IFIP* 1996.

[17] N. Robertson, P.D. Seymour, "Graph Minors XIII. The disjoint paths problem," *J. Combinatorial Theory Ser. B* 63(1995), pp. 65–110.

[18] R. Tijdeman, "On a telephone problem," *Nieuw Arch. Wisk.* 19(1971), pp. 188–192.