Temporal Network Optimization Subject to Connectivity Constraints

Paul G. Spirakis

joint work with
George B. Mertzios
Othon Michail
Ioannis Chatzigiannakis

Computer Technology Institute & Press "Diophantus" (CTI), Greece
Department of Computer Science, University of Liverpool, UK
School of Engineering and Computing Sciences, Durham University, UK

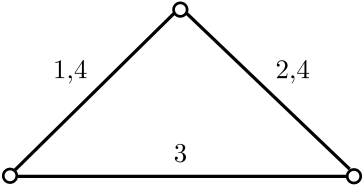
40th International Colloquium on Automata, Languages and Programming (ICALP)
July 8-12, 2013, Riga, Latvia

Temporal Networks

Definition (Temporal Graph)

Let $G = (V, E)$ be a (di)graph and $\lambda : E \to 2^{\mathbb{N}}$ be a labeling of G. Then $\lambda(G)$ is the temporal graph (or dynamic graph) of G w.r.t. λ.
Furthermore, G is the underlying graph of $\lambda(G)$.

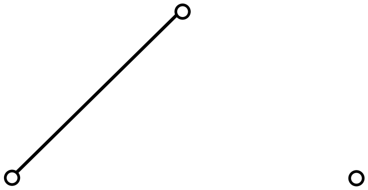
- Loosely speaking a network that changes with time
- Labels indicate availability times of edges



Definition (Temporal Graph)

Let $G = (V, E)$ be a (di)graph and $\lambda : E \to 2^{\mathbb{N}}$ be a labeling of G. Then $\lambda(G)$ is the temporal graph (or dynamic graph) of G w.r.t. λ. Furthermore, G is the underlying graph of $\lambda(G)$.

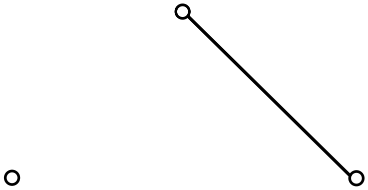
- Loosely speaking a network that changes with time
- Labels indicate availability times of edges



Definition (Temporal Graph)

Let $G = (V, E)$ be a (di)graph and $\lambda : E \to 2^{\mathbb{N}}$ be a labeling of G. Then $\lambda(G)$ is the temporal graph (or dynamic graph) of G w.r.t. λ. Furthermore, G is the underlying graph of $\lambda(G)$.

- Loosely speaking a network that changes with time
- Labels indicate availability times of edges

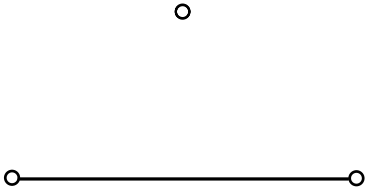


Temporal Networks

Definition (Temporal Graph)

Let $G = (V, E)$ be a (di)graph and $\lambda : E \to 2^{\mathbb{N}}$ be a labeling of G. Then $\lambda(G)$ is the temporal graph (or dynamic graph) of G w.r.t. λ. Furthermore, G is the underlying graph of $\lambda(G)$.

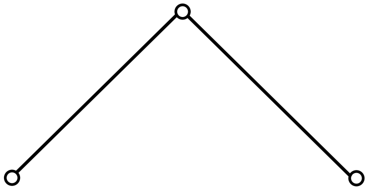
- Loosely speaking a network that changes with time
- Labels indicate availability times of edges



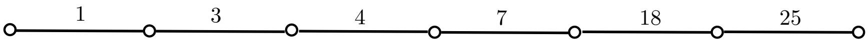
Definition (Temporal Graph)

Let $G = (V, E)$ be a (di)graph and $\lambda : E \to 2^{\mathbb{N}}$ be a labeling of G. Then $\lambda(G)$ is the temporal graph (or dynamic graph) of G w.r.t. λ. Furthermore, G is the underlying graph of $\lambda(G)$.

- Loosely speaking a network that changes with time
- Labels indicate availability times of edges

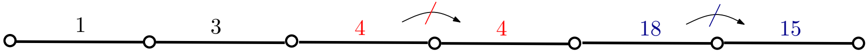


- Paths with strictly increasing labels
- a.k.a. journeys
- A journey:



Time-Respecting Paths

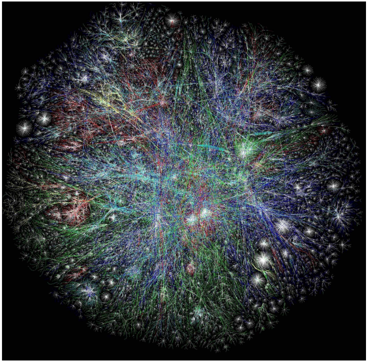
- Paths with strictly increasing labels
- a.k.a. journeys
- A non-journey:



Motivation

A great variety of systems are dynamic:

- Modern communication networks: inherently dynamic, dynamicity may be of high rate
  - mobile ad hoc, sensor, peer-to-peer, opportunistic, and delay-tolerant networks
- Social networks: social relationships between individuals change, existing individuals leave, new individuals enter



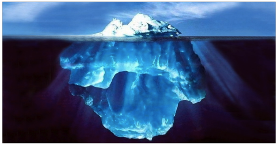
- Transportation networks: transportation units change their positions in the network as time passes
- Physical systems: e.g. systems of interacting particles

State of the Art

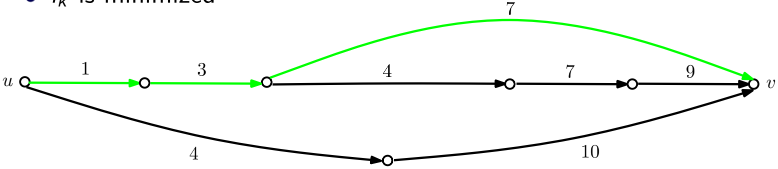
- Traditional communication networks: topology modifications are rare
- The structural and algorithmic properties of temporal graphs are not well understood yet
- Single-label temporal graphs
  - The max-flow min-cut theorem holds with unit capacities [Berman, '96]
  - Menger's theorem is violated [Kempe, Kleinberg, Kumar, STOC, '00]
- Continuous availabilities (intervals)
  - natural model but different techniques (e.g. journey problems [Xuan *et al.*, IJFCS, '03], dynamic flows [Fleischer, Tardos, Op. Res. Let., '98])
- Distributed Computing on Dynamic Networks
  - Worst-case dynamicity [Kuhn, Lynch, Oshman, STOC, '10], [Michail, Chatzigiannakis, Spirakis, JPDC, '13]
  - Population Protocols (interacting automata) [Angluin *et al.*, Distr. Comp., '06], [Michail, Chatzigiannakis, Spirakis, Book, '11]
  - Randomly Dynamic Networks [Clementi *et al.*, PODC, '08]

Overview-Contribution

- We give efficient algorithms for shortest journeys
- We state a temporal analogue of Menger's theorem and prove it valid for arbitrary temporal graphs
- We define cost minimization parameters for temporal network design
  - temporality, temporal cost, and age
  - satisfy some connectivity property: all paths and all reachabilities
  - We provide upper and lower bounds for basic graph families, e.g. rings, DAGs, trees, and a trade-off between temporality and age
  - We give a generic method for lower-bounding the temporality
  - **APX**-hardness result for temporal cost and an approximation algorithm
- Just the tip of the iceberg...



- Foremost (u, v)-journey $(e_1, l_1), (e_2, l_2), \ldots, (e_k, l_k)$ from time t
  - $l_1 \geq t$ and
  - l_k is minimized



Theorem

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

- $\alpha(\lambda) = \lambda_{\max} - \lambda_{\min} + 1$: the age of a temporal graph

Journey Problems

- Weighted temporal graph: In addition to λ a positive weight $w(e)$ is assigned to every $e \in E$
- Shortest Journey: Minimizes the sum of the weights of its edges
- Let $\lambda(G)$ be a weighted single-label temporal graph (i.e. $|\lambda(e)| = 1$ for all $e \in E$)

Theorem

For any two nodes $s, v \in V$, we can compute a shortest journey between s and v in $\lambda(G)$ (or report that no such journey exists) in $O(m \log m + \sum_{u \in V} \delta_u^2) = O(n^3)$ time.

- δ_u is the degree of node u, $m = |E|$

Menger's Analogue for Temporal Graphs

Theorem (Menger's theorem)

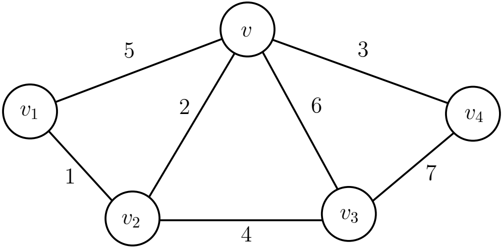
The maximum number of node-disjoint s-v paths is equal to the minimum number of nodes needed to separate s from v.

- Does it carry over to temporal graphs?
- Previous known result was negative

Theorem (Kempe, Kleinberg, Kumar, STOC '00)

There is no analogue of Menger's theorem, at least in its original formulation, for arbitrary single-label temporal networks.

- A violation of Menger's theorem



- 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

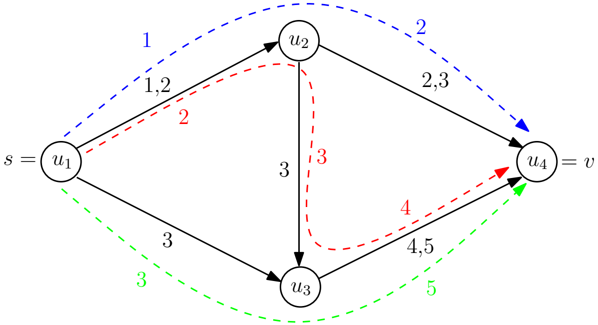
- We give a positive result
  - An analogue of Menger's theorem valid for all temporal networks
- Two journeys are out-disjoint if they never leave from the same node at the same time
- Remove departure time t from node u:
  - for all edges (u, w), remove label t from (u, w) (if it exists)

Theorem

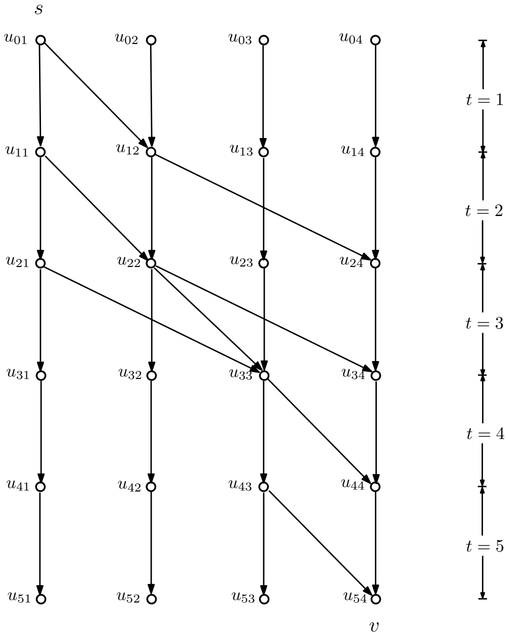
Take any temporal graph $\lambda(G)$ with two distinguished nodes s and v. The maximum number of out-disjoint journeys from s to v is equal to the minimum number of node departure times needed to separate s from v.

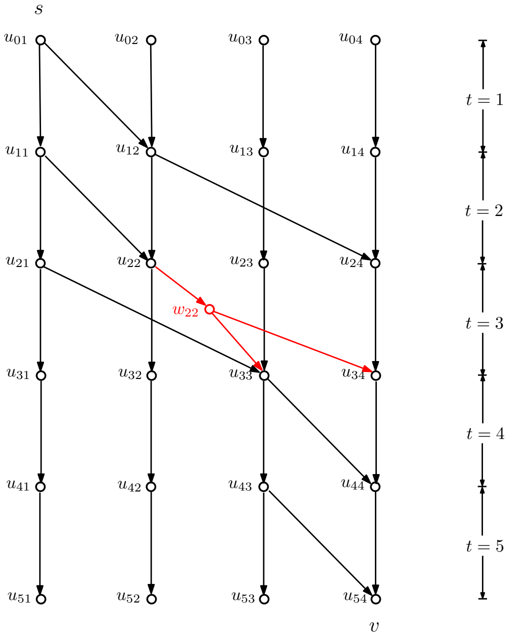
Menger's Analogue: An Example

- 3 out-disjoint journeys from s to v

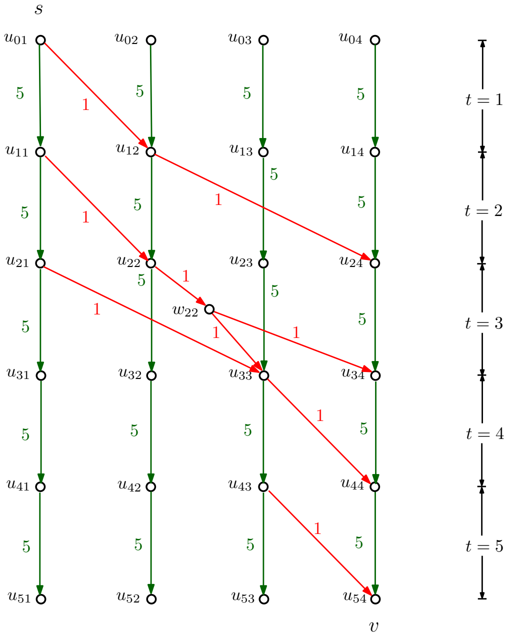


Menger's Analogue: An Example

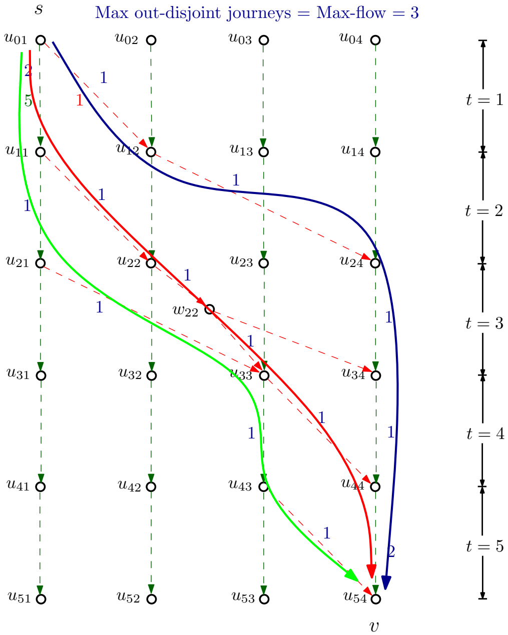



s
u_{01}
u_{02}
u_{03}
u_{04}
u_{11}
u_{12}
u_{13}
u_{14}
u_{21}
u_{22}
u_{23}
u_{24}
w_{22}
u_{31}
u_{32}
u_{33}
u_{34}
u_{41}
u_{42}
u_{43}
u_{44}
u_{51}
u_{52}
u_{53}
u_{54}
v
$t = 1$
$t = 2$
$t = 3$
$t = 4$
$t = 5$

Menger's Analogue: An Example



Menger's Analogue: An Example



Menger's Analogue: Distributed Token Gathering

- Distributed Model
  - $\lambda(G, t)$ is connected at all times $t \in \mathbb{N}$
  - $k \leq n$ tokens assigned to some given source nodes $S \subseteq V$
  - In each (discrete, synchronous) round i, each node broadcasts a single token to all its current neighbors (i.e. those defined by $E(i)$)

Lemma (Dutta *et al.*, SODA, '13)

All the tokens can be sent to any given v in $O(n)$ rounds.

- We substantially simplify the proof via Menger's Temporal Analogue.
- Given a mapping $N : S \to \mathbb{N}_{\geq 1}$ so that $\sum_{s \in S} N(s) = k$, we prove:

Lemma

Let the age be $\alpha(\lambda) = n + k$. There are at least k out-disjoint journeys from S to any given v such that $N(s_i)$ journeys leave from each source node s_i.

Cost Minimization Parameters

- (Di)graph $G = (V, E)$, $\alpha_{\max} \in \mathbb{N}$: upper bound on the age, Connectivity property $\mathcal{P}$

Definition (Temporality)

The temporality of $(G, \mathcal{P}, \alpha_{\max})$ is

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

- i.e. minimize the maximum number of labels of an edge while satisfying $\mathcal{P}$ and having age at most $\alpha_{\max}$

Definition (Temporal Cost)

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

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

- i.e. minimize the total number of labels used

Cost Minimization Parameters

- Similarly we define the age optimization criterion
- $\tau_{\max} \in \mathbb{N}$: upper bound on the temporality

Definition (Age)

The age of $(G, \mathcal{P}, \tau_{\max})$ is

$$\alpha(G, \mathcal{P}, \tau_{\max}) = \min_{\lambda \in \mathcal{P} \cap \mathcal{L}_{G, \tau_{\max}}} \alpha(\lambda)$$

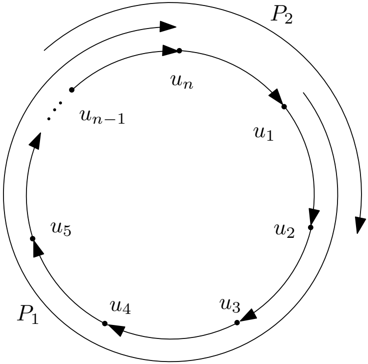
- i.e. minimize the age while satisfying $\mathcal{P}$ and having temporality at most $\tau_{\max}$
- Minimizing such parameters is crucial for many real networks
  - Establishing and maintaining a connection does not come for free
  - e.g. in WSNs cost of edges is directly related to: power consumption of keeping nodes awake, broadcasting, listening, resolving collisions

Connectivity Properties

- λ preserves path P if it gives a journey on P
- We investigate the following connectivity properties
  - 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 u in G, λ preserves at least one simple path from u to $v\}$.
- Example
  - Given: directed ring $R = u_1, u_2, \ldots, u_n$
  - Problem: determine $\tau(R, all\ paths)$, i.e. the temporality of the ring subject to the all paths property (no constraint on the age here)
  - i.e. find a labeling λ that (i) preserves every simple path of the ring and (ii) at the same time minimizes the maximum number of labels of an edge

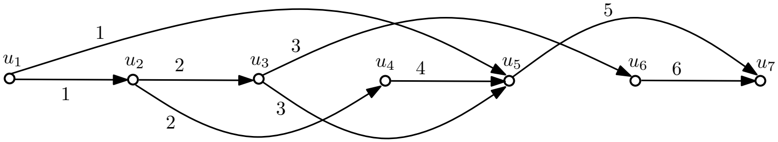
Ring Temporality Subject to All Paths

- Increasing labels on P_1 $\Rightarrow$ decreasing labels on (u_{n-1}, u_n) and (u_1, u_2)
- But P_2 uses first (u_{n-1}, u_n) and then (u_1, u_2) thus requires an increasing pair of labels on these edges
- To preserve both P_1, P_2 must use 2 labels on at least one of these two edges $\Rightarrow \tau(R, \textit{all paths}) \geq 2$



- The labeling that assigns to each edge (u_i, u_{i+1}) the labels $\{i, n+i\}$ preserves all simple paths, i.e. $\tau(R, \textit{all paths}) \leq 2$
- Conclusion: $\tau(R, \textit{all paths}) = 2$

Preserving All Paths of a DAG



Proposition

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

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

□

- It is sufficient to understand how $\tau(G, reach)$, behaves on strongly connected digraphs
- $\mathcal{C}(G)$: the set of all strongly connected components of a digraph G

Lemma

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

- Using this we prove that

Theorem (Generic Upper Bound)

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

- i.e. we can preserve all reachabilities of any digraph by using at most 2 labels on every edge

Restricting the Age

Theorem

If T is an undirected tree then $\tau(T, all\ paths, d(T)) \leq 2$.

- the age above is restricted to be at most the diameter $d(T)$ of T

Theorem (Age-Temporality Trade-off)

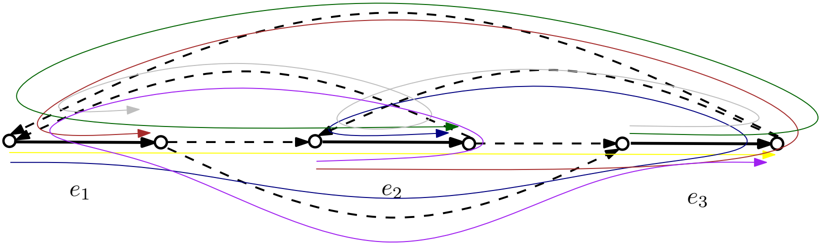
If G is a directed ring and $\alpha = (n-1) + k$, where $1 \leq k \leq n-1$, then

$$\tau(G, all\ paths, \alpha) = \Theta(n/k)$$

- *In particular,* $\lfloor \frac{n-1}{k+1} \rfloor + 1 \leq \tau(G, all\ paths, \alpha) \leq \lceil \frac{n}{k+1} \rceil + 1$
- *Moreover,* $\tau(G, all\ paths, n-1) = n-1$ *(i.e. when $k = 0$)*

Definition (Edge-kernel)

$K = \{e_1, e_2, \ldots, e_k\} \subseteq E(G)$ is an edge-kernel of G 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.



Generic Method for Lower Bounding Temporality

Theorem (Edge-kernel Lower Bound)

If a digraph G contains an edge-kernel of size k then $\tau(G, \text{all paths}) \geq k$.

Proof.

- $K = \{e_1, e_2, \ldots, e_k\}$: an edge-kernel of size k
- On every e_i sort the labels in an ascending order. $\lambda_l(e)$: the lth smallest label of edge e, e.g. $\lambda(e) = \{1, 3, 7\} \Rightarrow \lambda_1(e) = 1$, $\lambda_2(e) = 3$, $\lambda_3(e) = 7$
- Construct a permutation $\pi = (e_{j_1}, e_{j_2}, \ldots, e_{j_k})$ of K. e_{j_1}: edge with max λ_1, e_{j_2}: edge with max λ_2 between the remaining edges, ...
- Observe that π satisfies $\lambda_i(e_{j_i}) \geq \lambda_i(e_{j_{i+1}})$ for all $1 \leq i \leq k-1$
- π cannot use the labels $\lambda_1, \ldots, \lambda_{i-1}$ at edge e_{j_i} thus at edge e_{j_k} it can use none of the $k-1$ available labels $\Rightarrow$ needs a kth label $\square$

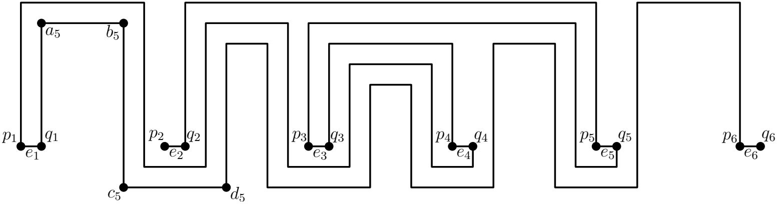
Applying the Edge-kernel Lower Bound

Lemma

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

Lemma

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



Hardness of Approximating the Cost

- Max-XOR: given a 2-CNF formula ϕ, max number of clauses of ϕ simultaneously XOR-satisfied in a truth assignment
- Max-XOR(k): every literal appears in at most k clauses of ϕ

Lemma

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

Theorem

There exists a truth assignment of ϕ XOR-satisfying at least k clauses iff $\kappa(G_\phi, reach, d(G_\phi)) \leq 39n - 4m - 2k$.

Theorem (Hardness of Approximating the Temporal Cost)

*Computing $\kappa(G, reach, d(G))$ is **APX**-hard, even when the maximum length of a directed cycle in G is 2 (i.e. very close to a DAG).*

Approximating the Cost

- $r(u) = |\{v \in V : v \text{ is reachable from } u\}|$
- $r(G) = \sum_{u \in V} r(u)$: total number of reachabilities in G

Theorem

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

Proof.

- $\text{OPT} \geq n - 1$
- Consider the following algorithm producing a labeling λ:
  - For all $u \in V$, compute a BFS out-tree T_u rooted at u
  - For all T_u, give to each edge at distance i from the root label i
- Maximum label used by λ is $d(G)$ and
- $\text{ALG} = |\lambda| = r(G)$: for each u, we label precisely $r(u)$ edges in T_u $\square$

Research Directions: To name a few...

- Still many interesting graph families to be investigated like regular or bounded-degree graphs
- Are there are other structural properties of G that cause a growth of temporality? (apart from edge-kernels)
- Other natural connectivity properties subject to which optimization is to be performed
  - e.g. preserve a shortest path between every reachable pair
  - depart from paths and require the preservation of more complex subgraphs
- Set the optimization criterion to be the age of λ
  - $\alpha(G, \text{all paths})$ is **NP**-hard (reduction from HAMPATH)
  - 2-factor approximation algorithm for $\alpha(G, \textit{reach}, 2)$

Research Directions: To name a few...

- Great room for approximation and randomized algorithms for all combinations of optimization parameters and connectivity constraints
- Polynomial-time algorithms for specific “easy to handle” graph families
- Consider periodic or probabilistic models of temporal graphs
- 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?

Thank You!