Non-Strict Temporal Exploration

Thomas Erlebach and **Jakob T. Spooner**

School of Informatics, University of Leicester

(Slides mostly prepared by Jakob)

Algorithmic Aspects of Temporal Graphs III
(Satellite Workshop of ICALP 2020)

July 7, 2020

Previous Work: Strict Temporal Exploration

Definition (Temporal graph $\mathcal{G}$)

Temporal graph $\mathcal{G} = \langle G_1, ..., G_L \rangle$:

- *underlying graph* G with N vertices
- sequence of static graphs $G_i \subseteq G$ with $V(G_i) = V(G)$ and $E(G_i) \subseteq E(G)$
- *time steps* $1 \leq i \leq L$, *lifetime* L

Strict temporal walk: Traverse at most one edge per time step.

Strict exploration schedule:

- Strict temporal walk W through $\mathcal{G}$ that visits all $v \in V(\mathcal{G})$.
- *Arrival time* of W: time step when last vertex is reached.

Strict Temporal Exploration Problem (TEXP)

Problem (Strict Temporal Exploration)

Input: Temporal graph $\mathcal{G}$, start vertex $s \in V(G)$.
Output: A strict exploration schedule starting from vertex s with earliest arrival time.

Typical assumptions:

- Full dynamic behaviour of $\mathcal{G}$ is known in advance
- Each G_i is connected, lifetime $L \geq N^2$.
  (Otherwise, NP-complete to decide if exploration schedule exists (Michail and Spirakis, 2014).)

STRICT TEXP: Some Known Results

- Introduced as TEXP by Michail and Spirakis (2014):
  - TEXP is NP-complete
  - TEXP admits an $O(D)$-approximation, where D is the temporal diameter ($D \leq N$)
- E, Hoffmann and Kammer (2015):
  - Worst-case exploration time is $\Theta(N^2)$
  - TEXP is $O(N^{1-\varepsilon})$-inapproximable.
- E, Kammer, Luo, Sajenko and Spooner (2019):
  - $O(d \cdot N^{1.75})$ steps suffice if each G_i has max degree $\leq d$
- The exploration time of various special classes of temporal graphs has also been studied.

Non-Strict Temporal Exploration

Non-Strict Temporal Graphs

- Allow an arbitrary number of edges to be crossed in the same time step.
- **Observation:** Only the connected components in each time step matter (for the exploration problem).

Definition (Non-strict temporal graph $\mathcal{G}$)

- $\mathcal{G} = \langle G_1, \ldots, G_L \rangle$ with vertex set V ($|V| = N$) and lifetime L
- Each G_i is a partition $\{C_{i,1}, \ldots, C_{i,s_i}\}$ of V

Non-Strict Temporal Walks

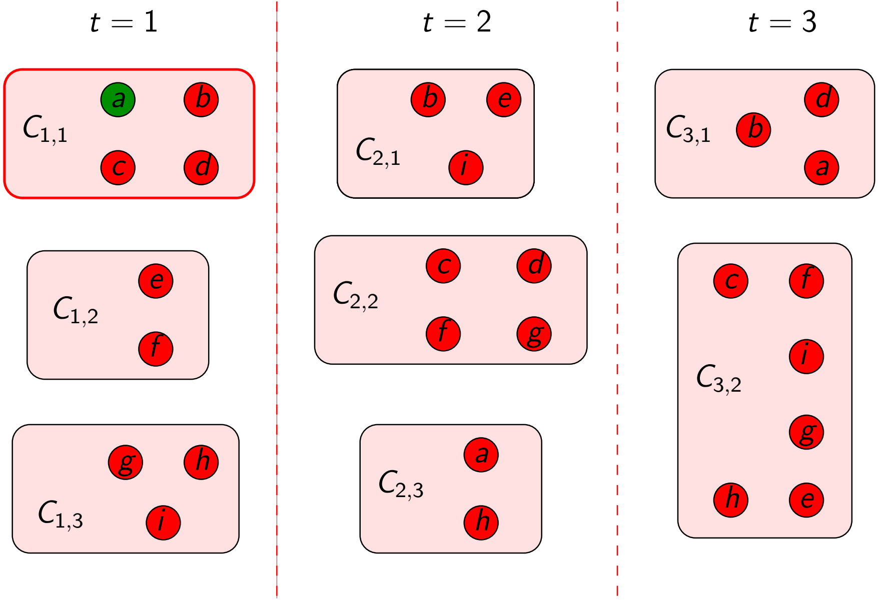
Definition (Non-strict temporal walk W)

A non-strict temporal walk W through a graph $\mathcal{G} = \langle G_1, \ldots, G_L \rangle$ is a length k-sequence of components $W = C_{1,j_1}, C_{2,j_2}, \ldots, C_{k,j_k}$ with $k \in [L]$, satisfying the following properties:

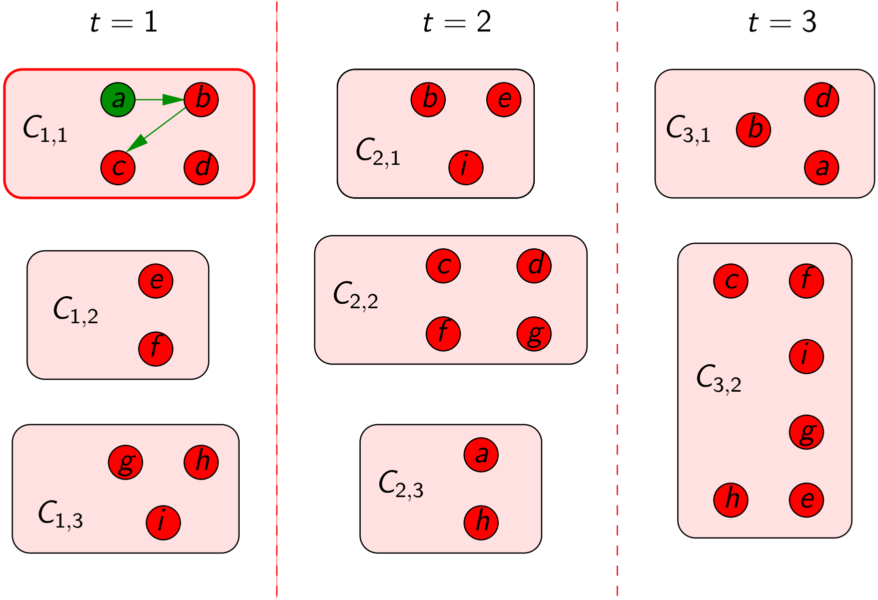
- For all $C_{i,j_i} \in W$ we have $C_{i,j_i} \in G_i$.
- Additionally, $C_{i,j_i} \cap C_{i+1,j_{i+1}} \neq \emptyset$ for all $i \in [k-1]$.

- A walk W **visits** all $v \in \bigcup_{i=1}^{k} C_{i,j_i}$.
- If $\bigcup_{i=1}^{k} C_{i,j_i} = V$ then W is a **non-strict exploration schedule**.

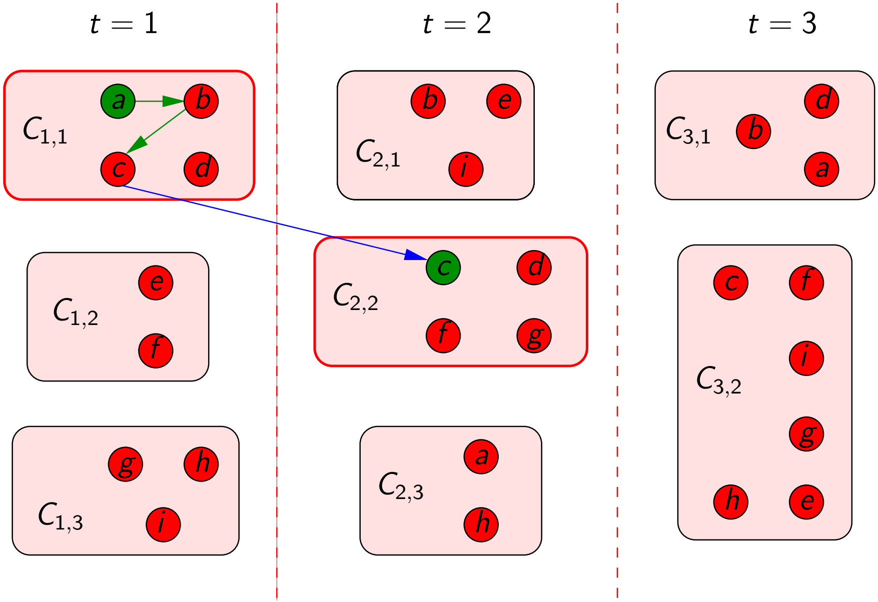
Example: Non-Strict Exploration Schedule



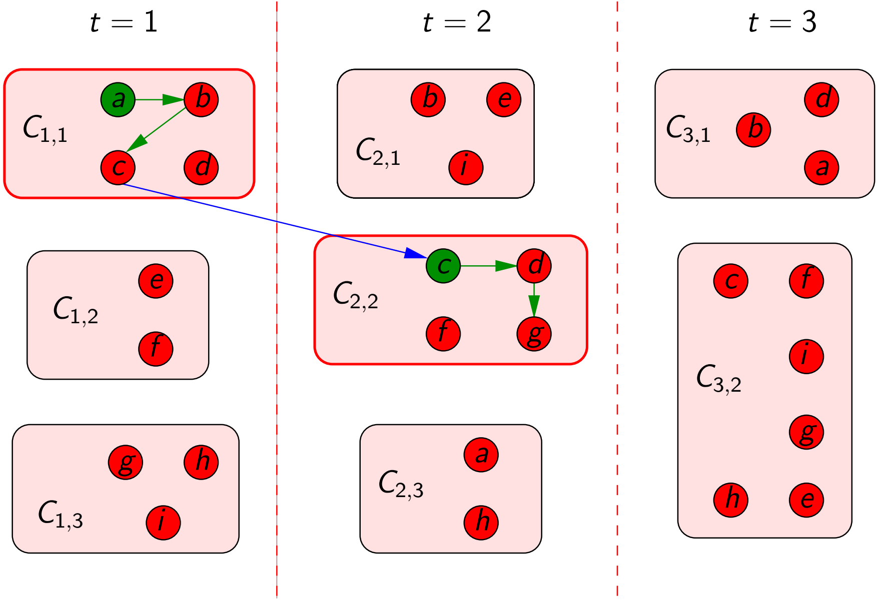
Example: Non-Strict Exploration Schedule



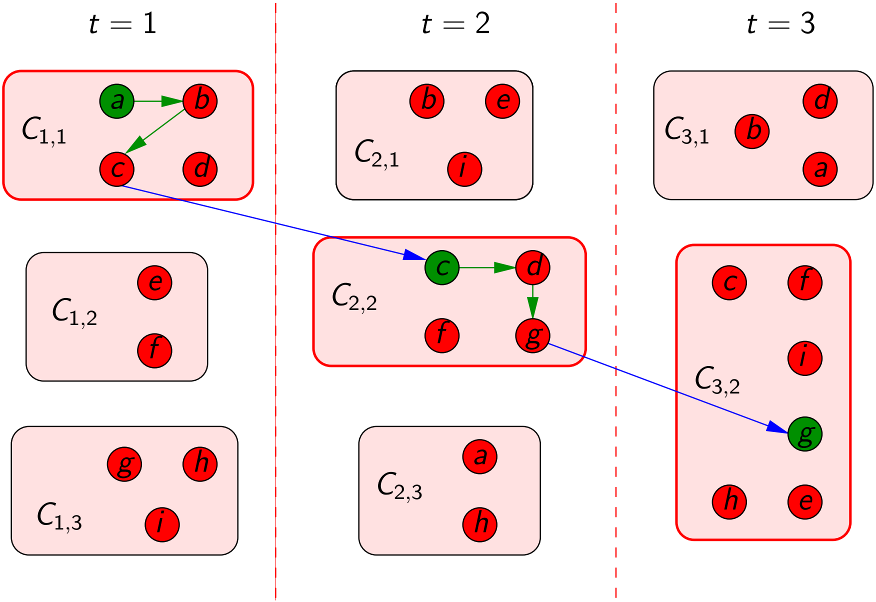
Example: Non-Strict Exploration Schedule



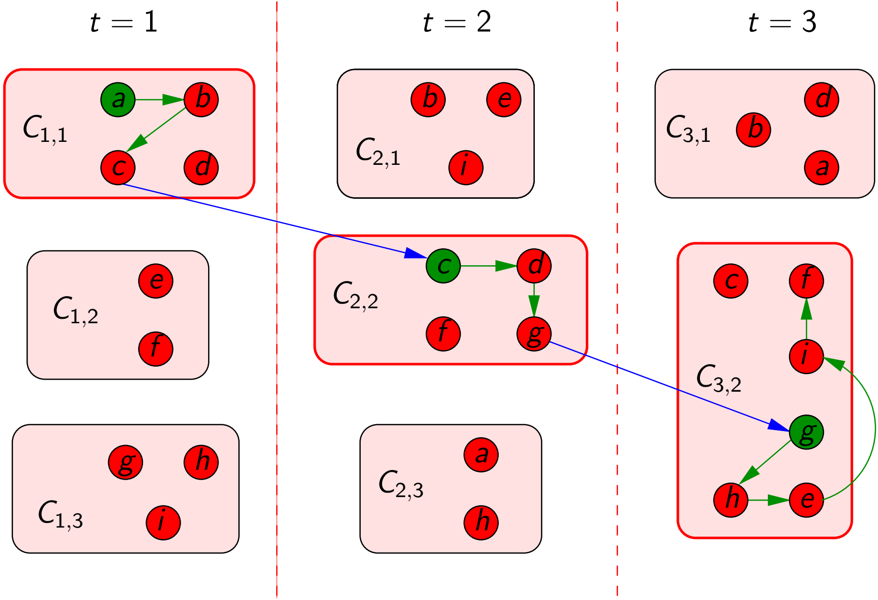
Example: Non-Strict Exploration Schedule



Example: Non-Strict Exploration Schedule



Example: Non-Strict Exploration Schedule



Previous work on non-strict temporal graphs

- Casteigts, Chaumette and Ferreira (2009) distinguish between strict/non-strict temporal journeys in the context of distributed algorithms.
- Barjon, Casteigts, Chaumette, Johnen and Neggaz (2014) describe algorithms for testing strict/non-strict temporal connectivity in sparse temporal graphs.
- Zschoche, Fluschnik, Molter and Niedermeier (2017) consider *temporal* (v, u)*-separators* in non-strict and strict setting.
- E, Kammer, Luo, Sajenko and Spooner (2019) prove arbitrary temporal graphs can be explored in $O(N^{1.75})$ time steps when up to 2 moves per step are allowed.

Deciding Non-Strict Temporal Exploration

- A non-strict temporal graph $\mathcal{G}$ does not necessarily admit an exploration schedule

Problem (Non-Strict TEXP Decision)

Input: A non-strict temporal graph $\mathcal{G}$ with lifetime L, and start vertex s.
Output: YES if $\mathcal{G}$ admits a non-strict exploration schedule W starting from s, and NO otherwise.

Deciding Non-Strict Temporal Exploration (cont.)

Theorem

Deciding NON-STRICT TEXP *is NP-complete.*

Deciding Non-Strict Temporal Exploration (cont.)

Theorem

Deciding NON-STRICT TEXP *is NP-complete.*

Proof sketch.

- Take arbitrary instance I of $3\mathrm{SAT}$ with n variables x_i ($i \in [n]$) and $m = O(n)$ clauses c_j.
- W.l.o.g., assume that no c_j contains both x_i and $\neg x_i$.
- Reduction: Construct non-strict temporal graph $\mathcal{G}$ such that:
  $\mathcal{G}$ admits exploration schedule $\iff$ I is satisfiable
  - For all $i \in [n]$, create 2 vertices v_i^T and v_i^F for variable x_i of I, m clause vertices c_j (one for each clause of I), and an additional vertex s.
  - Let the lifetime of $\mathcal{G}$ be $L = 2n$.

Proof of Theorem: Reducing 3SAT to NS-TEXP

Arrange vertices in components as follows (all unmentioned vertices in any step t are disconnected in that step):

$t = 1$: $\{s, v_1^T, v_1^F\}$

$t = 2$: $\{v_1^T\} \cup \{c_j : x_1 = 1 \text{ satisfies } c_j\}$ $\{v_1^F\} \cup \{c_j : x_1 = 0 \text{ satisfies } c_j\}$

...

$t = 2i - 1$: ($i \in [2, n]$) $\{v_{i-1}^T, v_{i-1}^F, v_i^T, v_i^F\}$

$t = 2i$: ($i \in [2, n]$) $\{v_i^T\} \cup \{c_j : x_i = 1 \text{ satisfies } c_j\}$ $\{v_i^F\} \cup \{c_j : x_i = 0 \text{ satisfies } c_j\}$

Proof of Theorem: I satisfiable $\iff$ $\mathcal{G}$ explorable

I satisfiable $\implies$ $\mathcal{G}$ admits exploration schedule W

Given satisfying assignment α, move in step $2i-1$ to v_i^T if $\alpha(x_i) = 1$ or to v_i^F otherwise. In step $2i$, explore all clause vertices satisfied by x_i in α.

Proof of Theorem: I satisfiable $\iff$ $\mathcal{G}$ explorable

I satisfiable $\implies$ $\mathcal{G}$ admits exploration schedule W

Given satisfying assignment α, move in step $2i-1$ to v_i^T if $\alpha(x_i)=1$ or to v_i^F otherwise. In step $2i$, explore all clause vertices satisfied by x_i in α.

$\mathcal{G}$ admits exploration schedule W $\implies$ I is satisfiable

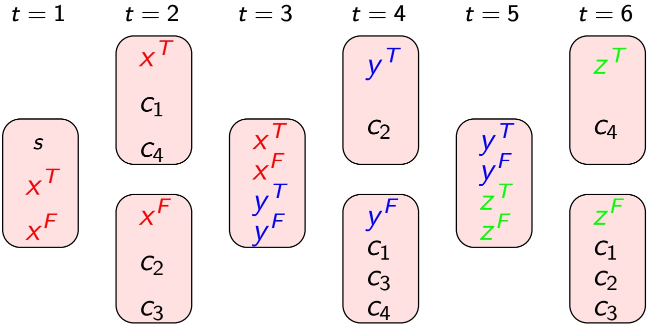
Each c_j can only be reached in a step $2i$ if it is contained in the true/false component of x_i. Since W visits all c_j, we can set $\alpha(x_i)=1$ or $\alpha(x_i)=0$ depending on the component visited in step $2i$ and obtain a satisfying assignment.

Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following NS-TEXP instance:

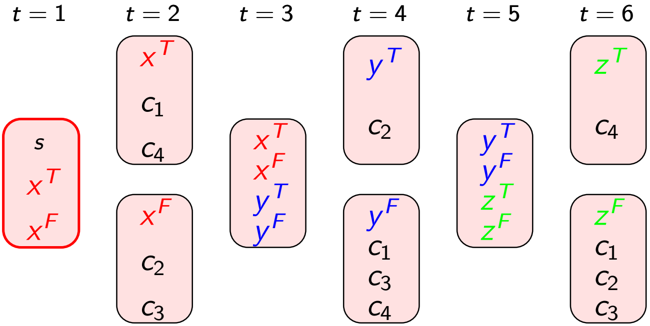


Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following NS-TEXP instance:

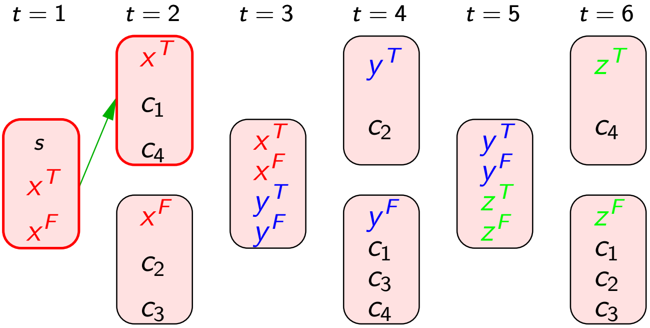


Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following $\mathrm{NS\text{-}TEXP}$ instance:

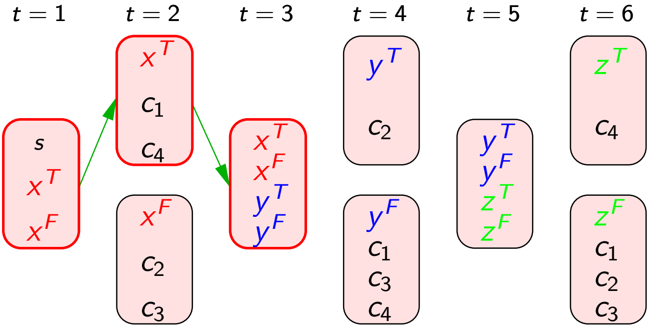


Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following NS-TEXP instance:

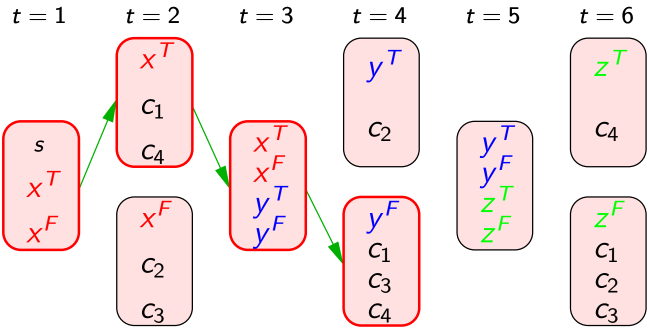


Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following NS-TEXP instance:

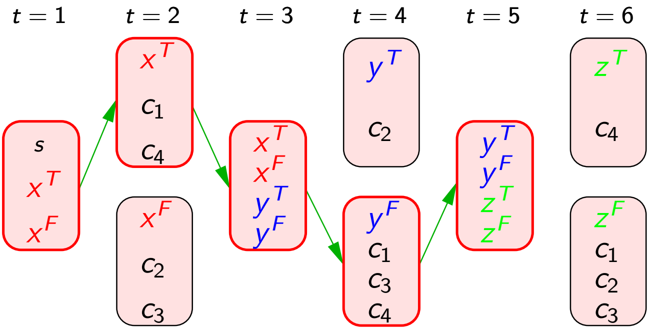


Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following NS-TEXP instance:

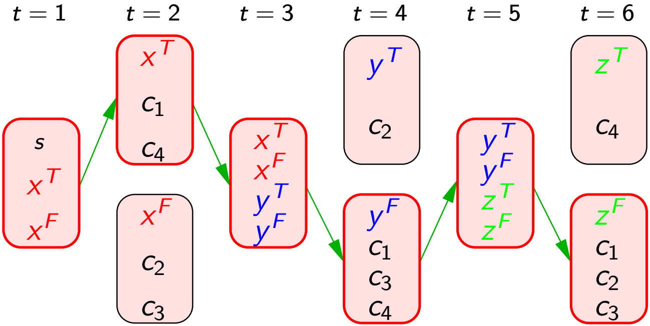


Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following NS-TEXP instance:

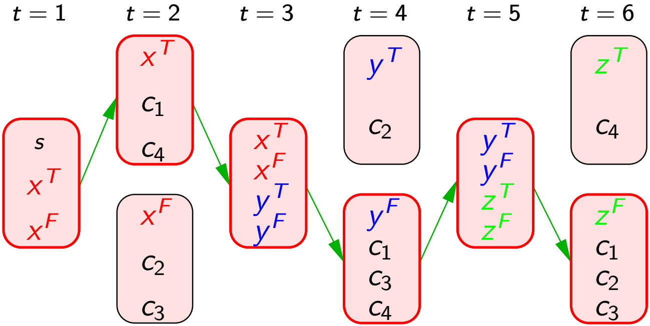


Proof of Theorem: Example

Consider the following 3CNF formula:

$$\phi = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee z)$$

Our reduction produces the following NS-TEXP instance:



There is a direct correspondence between the satisfying assignment $x = 1$, $y = 0$, $z = 0$ and the above exploration schedule.

Foremost NS-TEXP

- We are interested in assumptions that (together with large enough lifetime) **guarantee that non-strict exploration is possible**.

Foremost NS-TEXP

- We are interested in assumptions that (together with large enough lifetime) **guarantee that non-strict exploration is possible**.
- We consider the optimisation problem Foremost NS-TEXP for such instances: Find a **foremost** exploration schedule, i.e., one with earliest arrival time.

Foremost NS-TEXP

- We are interested in assumptions that (together with large enough lifetime) **guarantee that non-strict exploration is possible**.
- We consider the optimisation problem Foremost NS-TEXP for such instances: Find a **foremost** exploration schedule, i.e., one with earliest arrival time.

Assumption 1: Pairwise vertex-togetherness (PVT)

Every pair of vertices $u, v \in V(\mathcal{G})$ are contained in the same component at least once every $|V(\mathcal{G})| = N$ steps.

FOREMOST NS-TEXP

- We are interested in assumptions that (together with large enough lifetime) **guarantee that non-strict exploration is possible**.
- We consider the optimisation problem FOREMOST NS-TEXP for such instances: Find a **foremost** exploration schedule, i.e., one with earliest arrival time.

Assumption 1: Pairwise vertex-togetherness (PVT)

Every pair of vertices $u, v \in V(\mathcal{G})$ are contained in the same component at least once every $|V(\mathcal{G})| = N$ steps.

Observation: Under Assumption 1, any non-strict temporal graph $\mathcal{G}$ can be explored in N steps.

Approximation Hardness for Assumption 1

Theorem

FOREMOST NS-TEXP *is* $O(N^{1-\varepsilon})$*-inapproximable (unless P=NP) for input graphs satisfying the pairwise vertex-togetherness assumption*

Approximation Hardness for Assumption 1

Theorem

FOREMOST NS-TEXP *is* $O(N^{1-\varepsilon})$*-inapproximable (unless P=NP) for input graphs satisfying the pairwise vertex-togetherness assumption*

Proof sketch.

- Take instance of NS-TEXP obtained via the earlier reduction from 3SAT
- Add to the resulting graph $\mathcal{G}$ n^c *dummy vertices* d_k ($k \in [n^c]$), for some constant $c \geq 2$.
- $\mathcal{G}$ has lifetime $L = N = O(n^c)$.
- Components in steps $t \in [1, 2n]$ are arranged as in the earlier construction, with dummy vertices disconnected in all steps but $t = 1$, during which they are in the component containing s.
- During steps $t \in [2n + 1, N - 1]$, all vertices lie disconnected in $\mathcal{G}$; in step N all vertices lie in a single component.

Approximation Hardness for Assumption 1

Notice that if $\mathcal{G}$ cannot be explored by the end of $t = 2n$, then $N = \Theta(n^c)$ steps are required:

$t = 1$: $\{s, v_1^T, v_1^F, d_1, ..., d_{n^c}\}$

$t = 2$: $\{v_1^T\} \cup \{c_j : x_1 = 1 \text{ satisfies } c_j\}$ $\{v_1^F\} \cup \{c_j : x_1 = 0 \text{ satisfies } c_j\}$

...

$t = 2i - 1$: $(i \in [2, n])$ $\{v_{i-1}^T, v_{i-1}^F, v_i^T, v_i^F\}$

$t = 2i$: $(i \in [2, n])$ $\{v_i^T\} \cup \{c_j : x_i = 1 \text{ satisfies } c_j\}$ $\{v_i^F\} \cup \{c_j : x_i = 0 \text{ satisfies } c_j\}$

$t \in [2n + 1, N - 1]$: ... all vertices disconnected ...

$t = N$: $\{v : v \in V(\mathcal{G})\}$

Approximation Hardness for Assumption 1

Notice that if $\mathcal{G}$ cannot be explored by the end of $t = 2n$, then $N = \Theta(n^c)$ steps are required:

$t = 1$: $\{s, v_1^T, v_1^F, d_1, ..., d_{n^c}\}$

$t = 2$: $\{v_1^T\} \cup \{c_j : x_1 = 1 \text{ satisfies } c_j\}$ $\{v_1^F\} \cup \{c_j : x_1 = 0 \text{ satisfies } c_j\}$

...

$t = 2i - 1$: $(i \in [2, n])$ $\{v_{i-1}^T, v_{i-1}^F, v_i^T, v_i^F\}$

$t = 2i$: $(i \in [2, n])$ $\{v_i^T\} \cup \{c_j : x_i = 1 \text{ satisfies } c_j\}$ $\{v_i^F\} \cup \{c_j : x_i = 0 \text{ satisfies } c_j\}$

$t \in [2n + 1, N - 1]$: ... all vertices disconnected ...

$t = N$: $\{v : v \in V(\mathcal{G})\}$

Analysis: $\mathcal{G}$ can be explored in $2n$ steps iff I has a satisfying assignment, so deciding whether $\leq 2n$ or $\geq N$ are needed decides 3SAT instance I; the theorem follows for ratio $O(n^c/n) = O(N^{1-\varepsilon})$ where $\varepsilon = \frac{1}{c}$.

Assumption 2: Bounded Temporal Diameter

Definition (Temporal diameter of $\mathcal{G}$)

If every vertex can reach every other vertex within D steps (starting at any time $\leq L - D$), then $\mathcal{G}$ has temporal diameter D.

Assumption 2: Bounded Temporal Diameter

Definition (Temporal diameter of $\mathcal{G}$)

If every vertex can reach every other vertex within D steps (starting at any time $\leq L - D$), then $\mathcal{G}$ has temporal diameter D.

Assumption 2: Bounded temporal diameter

Input graph $\mathcal{G}$ has temporal diameter bounded by a constant c.

Assumption 2: Bounded Temporal Diameter

Definition (Temporal diameter of $\mathcal{G}$)

If every vertex can reach every other vertex within D steps (starting at any time $\leq L - D$), then $\mathcal{G}$ has temporal diameter D.

Assumption 2: Bounded temporal diameter

Input graph $\mathcal{G}$ has temporal diameter bounded by a constant c.

- Under Assumption 2, we can visit all vertices in arbitrary order in cN steps (actually, in $1 + (N-1)(c-1)$ steps).

Assumption 2: Bounded Temporal Diameter

Definition (Temporal diameter of $\mathcal{G}$)

If every vertex can reach every other vertex within D steps (starting at any time $\leq L - D$), then $\mathcal{G}$ has temporal diameter D.

Assumption 2: Bounded temporal diameter

Input graph $\mathcal{G}$ has temporal diameter bounded by a constant c.

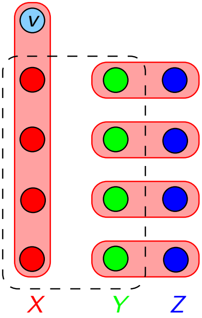
- Under Assumption 2, we can visit all vertices in arbitrary order in cN steps (actually, in $1 + (N-1)(c-1)$ steps).

We prove:

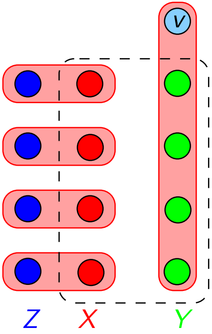
- Worst-case exploration time is $\Theta(N)$ when $c \geq 3$.
- Lower bound $\Omega(\sqrt{N})$ and upper bound $O(\sqrt{N} \log N)$ on worst-case exploration time when $c = 2$.

- Take $N = 3m + 1$ for some $m \geq 3$ and form 3 disjoint subsets X, Y and Z, each of size m. Arrange vertices as follows (red dashed lines indicate components):

Odd steps:

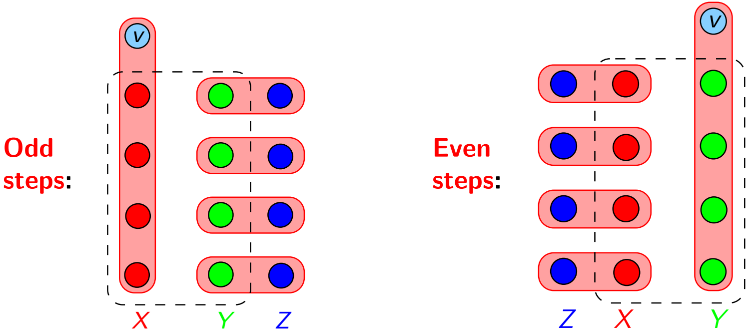


Even steps:



Lower Bound for Temporal Diameter $c = 3$

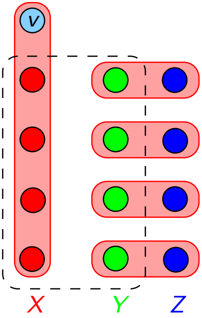
- Take $N = 3m + 1$ for some $m \geq 3$ and form 3 disjoint subsets X, Y and Z, each of size m. Arrange vertices as follows (red dashed lines indicate components):



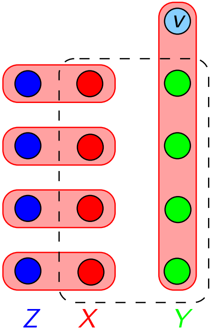
- Can check that ≤ 3 steps enough to reach any w from any u

- Take $N = 3m + 1$ for some $m \geq 3$ and form 3 disjoint subsets X, Y and Z, each of size m. Arrange vertices as follows (red dashed lines indicate components):

Odd steps:



Even steps:

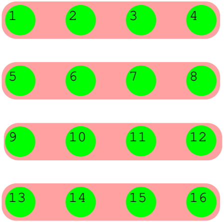


- Can check that ≤ 3 steps enough to reach any w from any u
- The vertices in Z need 3 steps to reach each other; repeating for all m gives $\Omega(N)$ time bound.

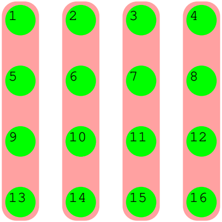
Lower Bound for Temporal Diameter $c = 2$

- Take $N = x^2$ for $x \geq 3$ and arrange vertices in x-by-x grid
- In odd steps, the components are rows of the grid, in even steps the components are columns:

Odd:

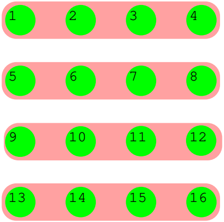


Even:

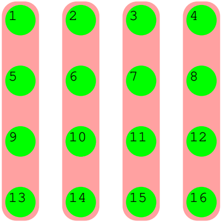


- Take $N = x^2$ for $x \geq 3$ and arrange vertices in x-by-x grid
- In odd steps, the components are rows of the grid, in even steps the components are columns:

Odd:



Even:

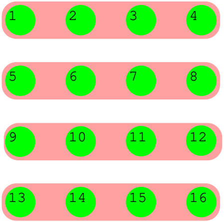


- In any pair of steps we can use one step to choose column, one to choose row $\implies$ $\mathcal{G}$ satisfies assumption

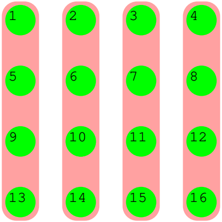
Lower Bound for Temporal Diameter $c = 2$

- Take $N = x^2$ for $x \geq 3$ and arrange vertices in x-by-x grid
- In odd steps, the components are rows of the grid, in even steps the components are columns:

Odd:



Even:



- In any pair of steps we can use one step to choose column, one to choose row $\implies \mathcal{G}$ satisfies assumption
- Any component contains exactly $\sqrt{N}$ vertices $\implies \Omega(\sqrt{N})$ steps required for exploration

Remark

These two lower bound constructions can be adapted to provide $O(N^{1-\varepsilon})$ and $O(N^{\frac{1}{2}-\varepsilon})$-inapproximability results in the $c \geq 3$ and $c = 2$ cases, respectively.

Upper Bound for Temporal Diameter $c = 2$

Theorem

Any temporal graph $\mathcal{G}$ that has temporal diameter $c = 2$ can be explored in $O(\sqrt{N} \log N)$ steps.

Upper Bound for Temporal Diameter $c = 2$

Theorem

Any temporal graph $\mathcal{G}$ that has temporal diameter $c = 2$ can be explored in $O(\sqrt{N} \log N)$ steps.

Proof outline.

Upper Bound for Temporal Diameter $c = 2$

Theorem

Any temporal graph $\mathcal{G}$ that has temporal diameter $c = 2$ can be explored in $O(\sqrt{N} \log N)$ steps.

Proof outline.

Claim In any pair of consecutive steps, at least one step has $\leq \sqrt{N}$ components

Upper Bound for Temporal Diameter $c = 2$

Theorem

Any temporal graph $\mathcal{G}$ that has temporal diameter $c = 2$ can be explored in $O(\sqrt{N} \log N)$ steps.

Proof outline.

Claim In any pair of consecutive steps, at least one step has $\leq \sqrt{N}$ components

- Construct walk in *blocks* of 3 steps; using Claim we are able to visit $\geq \frac{1}{\sqrt{N}}$ fraction of unvisited vertices in either 2nd or 3rd step of each block
- After k blocks the number of unvisited vertices is $\leq N \cdot (1 - \frac{1}{\sqrt{N}})^k$

Theorem

Any temporal graph $\mathcal{G}$ that has temporal diameter $c = 2$ can be explored in $O(\sqrt{N} \log N)$ steps.

Proof outline.

Claim In any pair of consecutive steps, at least one step has $\leq \sqrt{N}$ components

- Construct walk in *blocks* of 3 steps; using Claim we are able to visit $\geq \frac{1}{\sqrt{N}}$ fraction of unvisited vertices in either 2nd or 3rd step of each block
- After k blocks the number of unvisited vertices is $\leq N \cdot (1 - \frac{1}{\sqrt{N}})^k$
- Thus, $k \leq \sqrt{N} \log n$ blocks are enough to explore $\mathcal{G}$

Proof of Claim for Steps $t, t+1$

- If all components have size $> \sqrt{N}$ in step t, we are done. Otherwise, use this observation:

Proof of Claim for Steps $t, t+1$

- If all components have size $> \sqrt{N}$ in step t, we are done. Otherwise, use this observation:

Observation

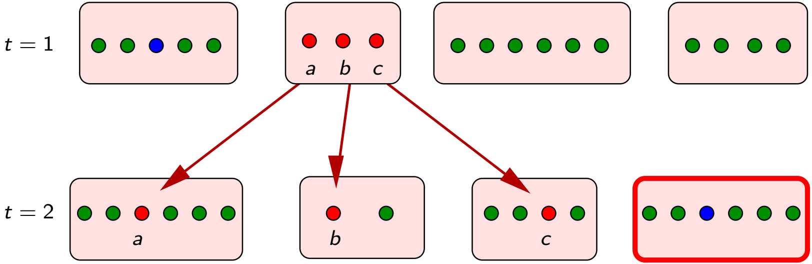
The number of components in step $t+1$ is upper bounded by the size of the smallest component in step t.

- If all components have size $> \sqrt{N}$ in step t, we are done. Otherwise, use this observation:

Observation

The number of components in step $t+1$ is upper bounded by the size of the smallest component in step t.

Proof sketch.



Conclusion

Our Results:

- Deciding if a temporal graph admits a non-strict exploration schedule is NP-complete
- Upper/lower bounds on worst-case exploration time under two assumptions (pairwise vertex-togetherness, bounded temporal diameter)
- FOREMOST NON-STRICT TEXP is hard to approximate under both assumptions

Conclusion

Our Results:

- Deciding if a temporal graph admits a non-strict exploration schedule is NP-complete
- Upper/lower bounds on worst-case exploration time under two assumptions (pairwise vertex-togetherness, bounded temporal diameter)
- FOREMOST NON-STRICT TEXP is hard to approximate under both assumptions

Open Questions:

- Close the $\Theta(\log n)$ gap for temporal diameter $c = 2$
- Analyse complexity/exploration time of FOREMOST NS-TEXP when the graph satisfies other assumptions that guarantee explorability

Thank you!

Any questions?