Parameterized Temporal Exploration Problems

Thomas Erlebach

Durham University

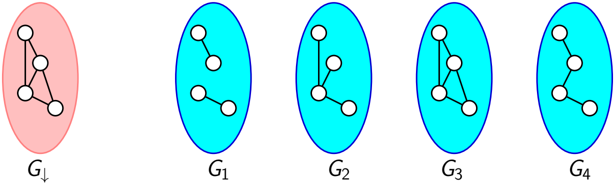
Joint work with Jakob T. Spooner

Algorithmic Aspects of Temporal Graphs V

ICALP 2022 – 4 July 2022

Definition (Temporal Graph)

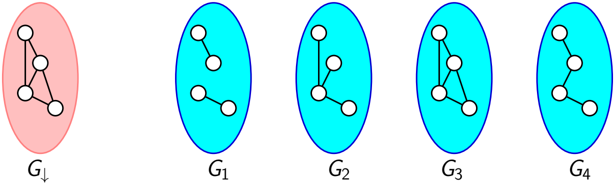
A temporal graph $\mathcal{G} = \langle G_1, ..., G_L \rangle$ with underlying graph $G_\downarrow = (V, E)$ and lifetime L consists of L static graphs (layers, steps) $G_i = (V, E_i)$ with $E_i \subseteq E$.



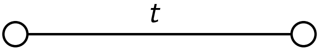
Temporal graphs

Definition (Temporal Graph)

A temporal graph $\mathcal{G} = \langle G_1, \ldots, G_L \rangle$ with underlying graph $G_\downarrow = (V, E)$ and lifetime L consists of L static graphs (layers, steps) $G_i = (V, E_i)$ with $E_i \subseteq E$.

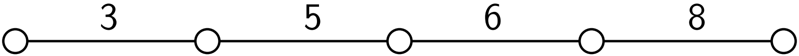


- If $e \in E_t$, we call (e, t) a time-edge.

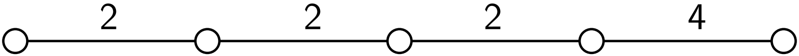


Temporal walks

- Strict temporal walk: increasing time steps



- Non-strict temporal walk: non-decreasing time steps



Temporal exploration

Temporal Exploration Problem (TEXP)

Given a temporal graph $\mathcal{G}$ and start vertex s, decide whether there is a temporal walk that starts at s at time 1 and visits all vertices.

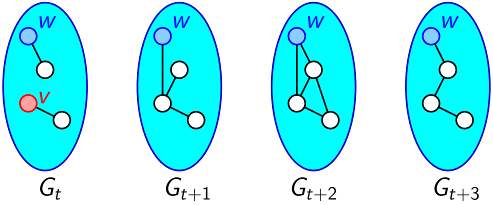
Previous work

- Strict TEXP:
  - Michail and Spirakis, 2014, 2016: *NP*-complete to decide if a temporal graph can be explored
  - If every G_t is connected and $L \geq n^2$:
    - Can be explored in $O(n^2)$ steps, no $O(n^{1-\varepsilon})$ approximation for Foremost-TEXP unless $P = NP$ (E, Hoffmann and Kammer, 2015, 2021).
    - Subquadratic upper bounds on exploration time for many special cases (EHK'15,'21; IW'18; EKLSS'19; . . .)

Previous work

- Strict TEXP:
  - Michail and Spirakis, 2014, 2016: NP-complete to decide if a temporal graph can be explored
  - If every G_t is connected and $L \geq n^2$:
    - Can be explored in $O(n^2)$ steps, no $O(n^{1-\varepsilon})$ approximation for Foremost-TEXP unless $P = NP$ (E, Hoffmann and Kammer, 2015, 2021).
    - Subquadratic upper bounds on exploration time for many special cases (EHK'15,'21; IW'18; EKLSS'19; ...)
- Non-strict TEXP (E and Spooner, 2020):
  - NP-complete to to decide if a temporal graph can be explored
  - Temporal diameter 2: $O(\sqrt{n} \cdot \log n)$ steps, no $O(n^{\frac{1}{2}-\varepsilon})$-approximation unless $P = NP$
  - Temporal diameter 3: may require up to $\Theta(n)$ steps, no $O(n^{1-\varepsilon})$-approximation unless $P = NP$

Auxiliary tool

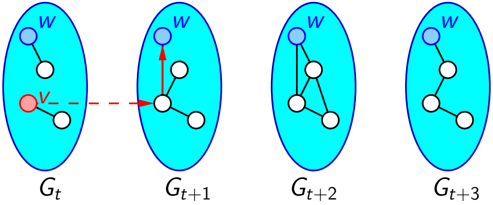


Algorithm for foremost temporal walk (cf. Bui-Xuan et al., 2013; Wu et al., 2014)

Given a temporal graph $\mathcal{G}$, a vertex v, and a time t, one can compute in $O(Ln^2)$ time a foremost (i.e., earliest arrival time) strict temporal walk to any (or all) destination vertices w starting at v at time t.

A similar algorithm exists for non-strict walks.

Auxiliary tool

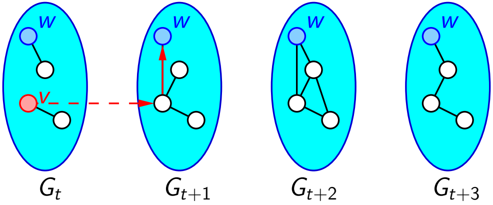


Algorithm for foremost temporal walk (cf. Bui-Xuan et al., 2013; Wu et al., 2014)

Given a temporal graph $\mathcal{G}$, a vertex v, and a time t, one can compute in $O(Ln^2)$ time a foremost (i.e., earliest arrival time) strict temporal walk to any (or all) destination vertices w starting at v at time t.

A similar algorithm exists for non-strict walks.

Auxiliary tool



Algorithm for foremost temporal walk (cf. Bui-Xuan et al., 2013; Wu et al., 2014)

Given a temporal graph $\mathcal{G}$, a vertex v, and a time t, one can compute in $O(Ln^2)$ time a foremost (i.e., earliest arrival time) strict temporal walk to any (or all) destination vertices w starting at v at time t.

A similar algorithm exists for non-strict walks.

The main difficulty in TEXP is to decide the best order in which vertices should be visited.

Parameterized complexity of TEXP

- TEXP is a computationally difficult problem.
- Are there efficient parameterized algorithms for (variants of) TEXP?

Parameterized complexity of TEXP

- TEXP is a computationally difficult problem.
- Are there efficient parameterized algorithms for (variants of) TEXP?

Definition

A problem is fixed-parameter tractable (FPT) if an instance of size n with parameter k can be solved in $f(k) \cdot n^{O(1)}$ time.

Parameterized complexity of TEXP

- TEXP is a computationally difficult problem.
- Are there efficient parameterized algorithms for (variants of) TEXP?

Definition

A problem is fixed-parameter tractable (FPT) if an instance of size n with parameter k can be solved in $f(k) \cdot n^{O(1)}$ time.

Examples of possible parameters for TEXP:

- L = lifetime
- γ = maximum number of connected components per layer

Variants of TEXP

k-fixed TEXP

Given a temporal graph $\mathcal{G}$ and start vertex s and vertex subset $X \subseteq V$ with $|X| = k$, decide whether there is a temporal walk that starts at s at time 1 and visits at least all vertices in X.

Variants of TEXP

k-fixed TEXP

Given a temporal graph $\mathcal{G}$ and start vertex s and vertex subset $X \subseteq V$ with $|X| = k$, decide whether there is a temporal walk that starts at s at time 1 and visits at least all vertices in X.

k-arbitrary TEXP

Given a temporal graph $\mathcal{G}$ and start vertex s and $k \in \mathbb{N}$, decide whether there is a temporal walk that starts at s at time 1 and visits k different vertices.

Variants of TEXP

k-fixed TEXP

Given a temporal graph $\mathcal{G}$ and start vertex s and vertex subset $X \subseteq V$ with $|X| = k$, decide whether there is a temporal walk that starts at s at time 1 and visits at least all vertices in X.

k-arbitrary TEXP

Given a temporal graph $\mathcal{G}$ and start vertex s and $k \in \mathbb{N}$, decide whether there is a temporal walk that starts at s at time 1 and visits k different vertices.

Set-TEXP

Given a temporal graph $\mathcal{G}$ and start vertex s and m vertex subsets $S_i \subseteq V$, is there a temporal walk that starts at s at time 1 and visits at least one vertex from each S_i.

Our results

Problem	Parameter	strict	non-strict
TEXP	L	FPT	**FPT**
TEXP	γ	NPC for $\gamma = 1$	**poly for** $\gamma = 1, 2$
k-fixed TEXP	k	**FPT**	FPT
k-arbitrary TEXP	k	**FPT**	FPT
Set-TEXP	L	W[2]-hard	**W[2]-hard**

Reminder:

- L = lifetime
- γ = maximum number of connected components per layer
- k = number of vertices to be visited

Results for strict temporal exploration problems

FPT algorithms for strict model

- k-fixed TEXP
  - Trivial: $O(k! \cdot n^2 L)$ (try all orders)

FPT algorithms for strict model

- k-fixed TEXP
  - Trivial: $O(k! \cdot n^2 L)$ (try all orders)
  - Use dynamic programming similar to the classical Held-Karp algorithm for TSP: $O(2^k k \cdot n^2 L)$ time

FPT algorithms for strict model

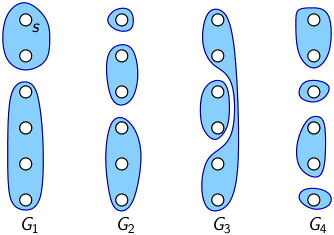
- k-fixed TEXP
  - Trivial: $O(k! \cdot n^2 L)$ (try all orders)
  - Use dynamic programming similar to the classical Held-Karp algorithm for TSP: $O(2^k k \cdot n^2 L)$ time
- k-arbitrary TEXP
  - Trivial: $O(n^k \cdot n^2 L)$ (try all vertex sequences of length k)

FPT algorithms for strict model

- k-fixed TEXP
  - Trivial: $O(k! \cdot n^2 L)$ (try all orders)
  - Use dynamic programming similar to the classical Held-Karp algorithm for TSP: $O(2^k k \cdot n^2 L)$ time
- k-arbitrary TEXP
  - Trivial: $O(n^k \cdot n^2 L)$ (try all vertex sequences of length k)
  - Use color coding (Alon, Yuster, Zwick 1995) and dynamic programming to get FPT algorithms:
    - Randomized $O((2e)^k L n^3 \log \frac{1}{\varepsilon})$ time, correct output with probability $1 - \varepsilon$
    - Derandomization: deterministic $(2e)^k k^{O(\log k)} L n^3 \log n$ time

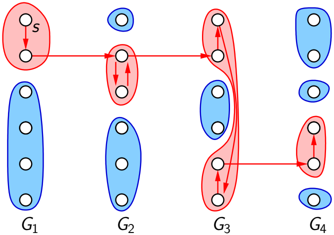
Results for non-strict temporal exploration problems

TEXP in the non-strict model



- A temporal walk can visit all vertices of its connected component in each time step.
- In step 1, the walk can visit all vertices in the connected component containing s.
- The connected components visited in two consecutive time steps must share a vertex.
- We can represent a temporal exploration by specifying the connected component that it visits in every step.

TEXP in the non-strict model



- A temporal walk can visit all vertices of its connected component in each time step.
- In step 1, the walk can visit all vertices in the connected component containing s.
- The connected components visited in two consecutive time steps must share a vertex.
- We can represent a temporal exploration by specifying the connected component that it visits in every step.

Non-strict TEXP with parameter L

- Bounded search tree algorithm

Non-strict TEXP with parameter L

- Bounded search tree algorithm
- Choices for the largest component visited by OPT:
  - As OPT visits all n vertices in L steps, it must visit one component containing at least n/L vertices.
  - In each of the L steps, there exist at most L components with at least n/L vertices.
  - $\Rightarrow$ There are $\leq L^2$ possible choices for the largest component visited by OPT.

Non-strict TEXP with parameter L

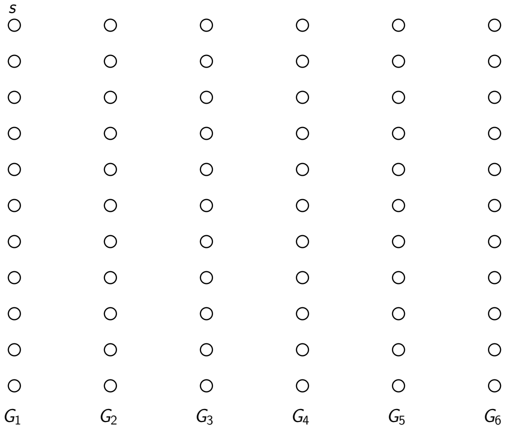
- Bounded search tree algorithm
- Choices for the largest component visited by OPT:
  - As OPT visits all n vertices in L steps, it must visit one component containing at least n/L vertices.
  - In each of the L steps, there exist at most L components with at least n/L vertices.
  - $\Rightarrow$ There are $\leq L^2$ possible choices for the largest component visited by OPT.
- If there are u unvisited vertices and $L - i$ steps left:
  - As OPT visits all u vertices in $L - i$ steps, it must visit one component containing at least $u/(L - i)$ vertices.
  - In each of the $L - i$ steps, there exist at most $L - i$ components with at least $u/(L - i)$ unvisited vertices.
  - $\Rightarrow$ There are $\leq (L - i)^2$ possible choices for the largest component visited by OPT.

Non-strict TEXP with parameter L

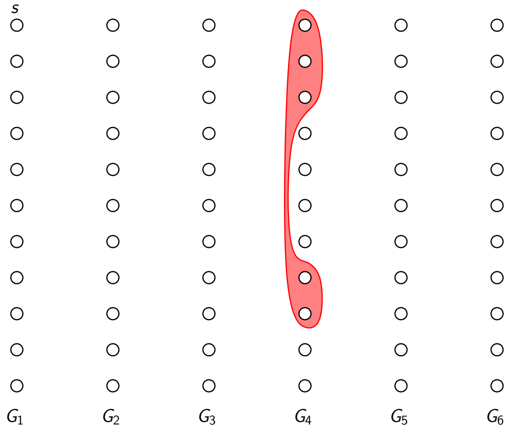
- Bounded search tree algorithm
- Choices for the largest component visited by OPT:
  - As OPT visits all n vertices in L steps, it must visit one component containing at least n/L vertices.
  - In each of the L steps, there exist at most L components with at least n/L vertices.
  - $\Rightarrow$ There are $\leq L^2$ possible choices for the largest component visited by OPT.
- If there are u unvisited vertices and $L - i$ steps left:
  - As OPT visits all u vertices in $L - i$ steps, it must visit one component containing at least $u/(L - i)$ vertices.
  - In each of the $L - i$ steps, there exist at most $L - i$ components with at least $u/(L - i)$ unvisited vertices.
  - $\Rightarrow$ There are $\leq (L - i)^2$ possible choices for the largest component visited by OPT.

Theorem

Non-strict TEXP can be solved in $O(L(L!)^2 n)$ time.

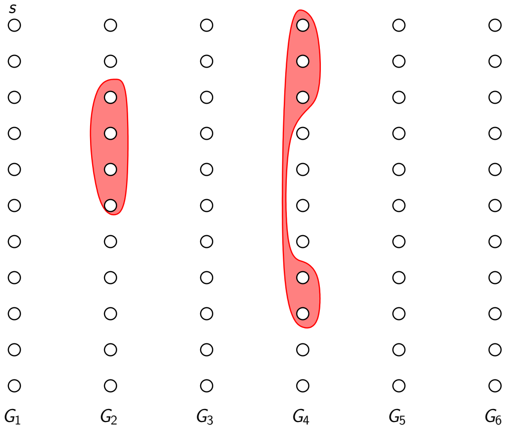


- For $i = 0, 1, \ldots,$ add a component in an unused step that has at least $u/(L - i)$ unvisited vertices
- When $u = 0$, complete the components into a temporal walk



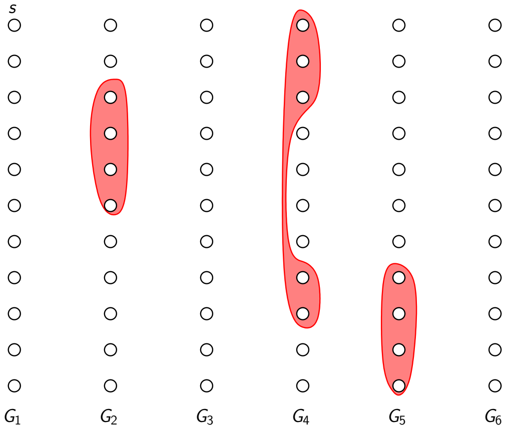
- For $i = 0, 1, \ldots$, add a component in an unused step that has at least $u/(L - i)$ unvisited vertices
- When $u = 0$, complete the components into a temporal walk

Illustration of the algorithm



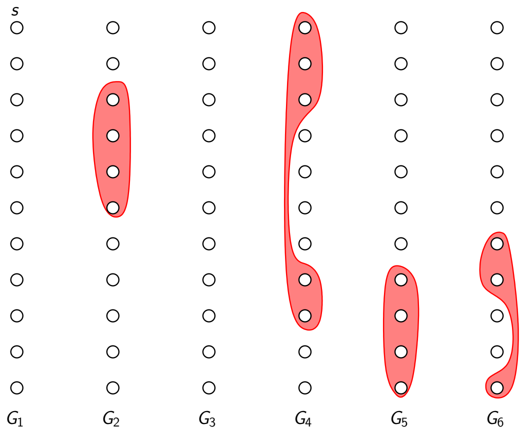
- For $i = 0, 1, \ldots,$ add a component in an unused step that has at least $u/(L - i)$ unvisited vertices
- When $u = 0$, complete the components into a temporal walk

Illustration of the algorithm



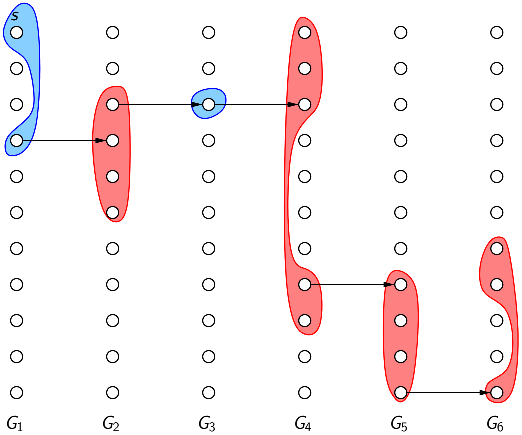
- For $i = 0, 1, \ldots$, add a component in an unused step that has at least $u/(L - i)$ unvisited vertices
- When $u = 0$, complete the components into a temporal walk

Illustration of the algorithm



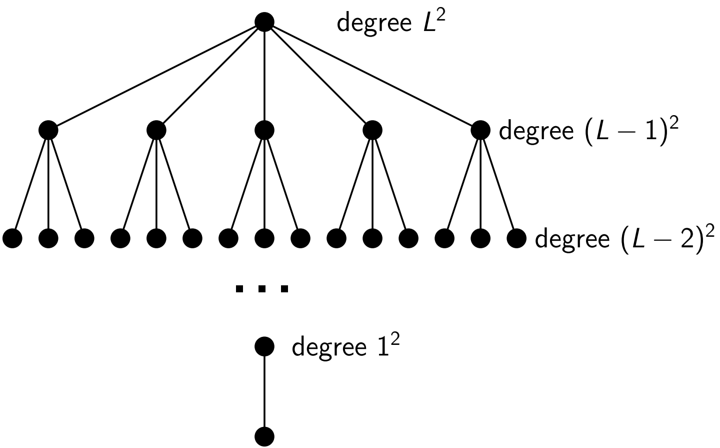
- For $i = 0, 1, \ldots,$ add a component in an unused step that has at least $u/(L - i)$ unvisited vertices
- When $u = 0$, complete the components into a temporal walk

Illustration of the algorithm



- For $i = 0, 1, \ldots$, add a component in an unused step that has at least $u/(L-i)$ unvisited vertices
- When $u = 0$, complete the components into a temporal walk

The resulting search tree



- Depth $\leq L$
- $O((L!)^2)$ nodes, time $O(nL)$ per node
- Total time $O(L(L!)^2 n)$

Non-strict TEXP with parameter γ

$\gamma =$ maximum number of connected components per layer

- With $\gamma = 1$ connected components per layer, non-strict exploration can trivially be completed in one step.

Non-strict TEXP with parameter γ

$\gamma =$ maximum number of connected components per layer

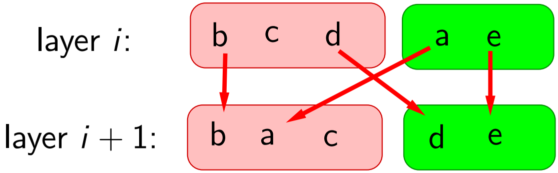
- With $\gamma = 1$ connected components per layer, non-strict exploration can trivially be completed in one step.
- Consider $\gamma = 2$. We can assume:
  - No two consecutive layers have identical connected components.
  - Every layer has exactly 2 connected components.

Possible transitions between consecutive layers

The transition from layer i to $i + 1$ can be one of two types:

The transition from layer i to $i+1$ can be one of two types:

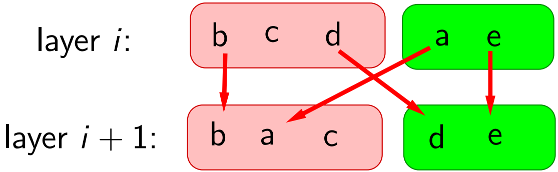
- **Free transition:** Each component of layer i can reach each component of layer $i+1$, for example:



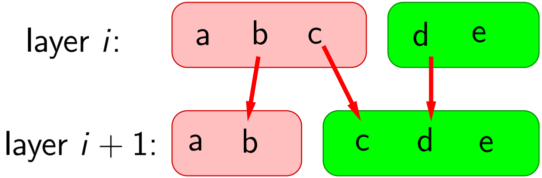
Possible transitions between consecutive layers

The transition from layer i to $i+1$ can be one of two types:

- **Free transition:** Each component of layer i can reach each component of layer $i+1$, for example:



- **Restricted transition:** One component of layer i cannot reach one component of layer $i+1$, for example:



Only possible if one component shrinks and the other grows.

Observation 1

Observation

If there is a restricted transition from layer i to $i + 1$, the whole graph can be explored from the shrinking component in layer i.

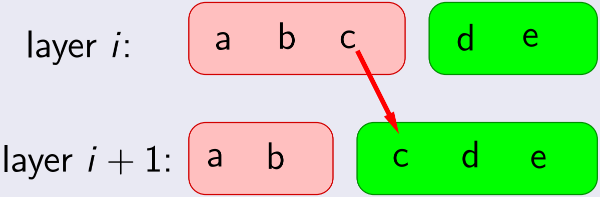
Observation 1

Observation

If there is a restricted transition from layer i to $i+1$, the whole graph can be explored from the shrinking component in layer i.

Proof.

Explore the shrinking component in step i, then the growing one in step $i+1$.



□

Observation 2

Observation

If a restricted transition follows a free transition, the whole graph can be explored.

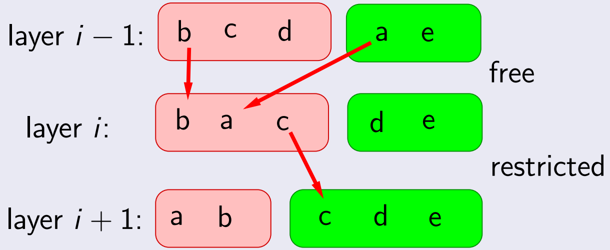
Observation 2

Observation

If a restricted transition follows a free transition, the whole graph can be explored.

Proof.

Move to the shrinking component in the free transition, then proceed as in the previous observation.



□

Observation 3

Observation

In $\log_2 n$ *consecutive free transitions, the whole graph can be explored.*

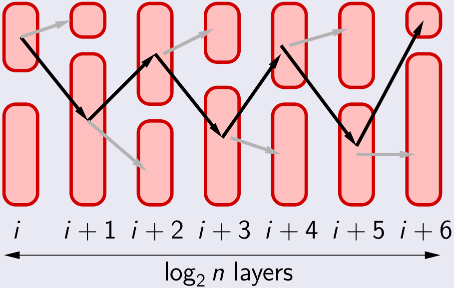
Observation 3

Observation

In $\log_2 n$ *consecutive free transitions, the whole graph can be explored.*

Proof.

We can visit at least half of the unvisited vertex in each step, so the exploration is finished after $\log_2 n$ steps.



Algorithm:

- If a restricted transition follows a free transition, answer YES.

Polynomial algorithm for $\gamma = 2$

Algorithm:

- If a restricted transition follows a free transition, answer YES.
- If there is an initial sequence of restricted transitions:
  - If s is always in the growing component, can only stay in that component.
  - If s is ever in the shrinking component, can complete the exploration.

Polynomial algorithm for $\gamma = 2$

Algorithm:

- If a restricted transition follows a free transition, answer YES.
- If there is an initial sequence of restricted transitions:
  - If s is always in the growing component, can only stay in that component.
  - If s is ever in the shrinking component, can complete the exploration.
- In the remaining steps with only free transitions, try the up to $2^{\log_2 n} = n$ possible choices for up to $\log_2 n$ steps.

Polynomial algorithm for $\gamma = 2$

Algorithm:

- If a restricted transition follows a free transition, answer YES.
- If there is an initial sequence of restricted transitions:
  - If s is always in the growing component, can only stay in that component.
  - If s is ever in the shrinking component, can complete the exploration.
- In the remaining steps with only free transitions, try the up to $2^{\log_2 n} = n$ possible choices for up to $\log_2 n$ steps.

Theorem

Non-strict TEXP with $\gamma = 2$ can be solved in $O(nL + n^2 \log n)$ time.

W[2]-hardness of Set-TEXP

Reminder:
Set-TEXP: Given m vertex subsets $S_i \subseteq V$, is there a temporal walk that starts at s and visits at least one vertex from each S_i?

Theorem

Set-TEXP with parameter L is W[2]-hard.

Reminder:
Set-TEXP: Given m vertex subsets $S_i \subseteq V$, is there a temporal walk that starts at s and visits at least one vertex from each S_i?

Theorem

Set-TEXP with parameter L is W[2]-hard.

Proof.

- Non-strict model: Parameterized reduction from SetCover
- Strict model: Parameterized reduction from HittingSet (works even if each G_i is a complete graph)

□

Conclusion

- **Our results**

Problem	Parameter	strict	non-strict
TEXP	L	FPT	**FPT**
TEXP	γ	NPC for $\gamma = 1$	**poly for** $\gamma = 1, 2$
k-fixed TEXP	k	**FPT**	FPT
k-arbitrary TEXP	k	**FPT**	FPT
Set-TEXP	L	W[2]-hard	**W[2]-hard**

- **Open problem**
  - Is non-strict TEXP with parameter γ (the maximum number of connected components per layer) in XP or even FPT?

Conclusion

- **Our results**

Problem	Parameter	strict	non-strict
TEXP	L	FPT	**FPT**
TEXP	γ	NPC for $\gamma = 1$	**poly for** $\gamma = 1, 2$
k-fixed TEXP	k	**FPT**	FPT
k-arbitrary TEXP	k	**FPT**	FPT
Set-TEXP	L	W[2]-hard	**W[2]-hard**

- **Open problem**
  - Is non-strict TEXP with parameter γ (the maximum number of connected components per layer) in XP or even FPT? Even for $\gamma = 3$ the complexity is open!

Thank you!

Questions?