Parameterized Algorithms for Multi-Label Periodic Temporal Graph Realization

Thomas Erlebach[1], Nils Morawietz[2], Petra Wolf[3]

[1] Durham University, UK
[2] Friedrich Schiller University Jena, Germany
[3] Université de Bordeaux, France

(slides (mostly) by Nils Morawietz)

AATG–VII, Tallinn, 7/7/24

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization
Input: A graph $G = (V, E)$, a duration matrix D, and a period Δ.
Question: Is there a temporal labeling $\lambda\colon E \to [1, \Delta]$ such that the Δ-periodic graph (G, λ) realizes D?

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization
Input: A graph $G = (V, E)$, a duration matrix D, and a period Δ.
Question: Is there a temporal labeling $\lambda \colon E \to [1, \Delta]$ such that the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5$

$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization
Input: A graph $G = (V, E)$, a duration matrix D, and a period Δ.
Question: Is there a temporal labeling $\lambda \colon E \to [1, \Delta]$ such that the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5$

$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization
Input: A graph $G = (V, E)$, a duration matrix D, and a period Δ.
Question: Is there a temporal labeling $\lambda \colon E \to [1, \Delta]$ such that the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5$

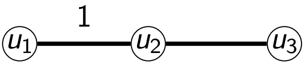
$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization
Input: A graph $G = (V, E)$, a duration matrix D, and a period Δ.
Question: Is there a temporal labeling $\lambda \colon E \to [1, \Delta]$ such that the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5$



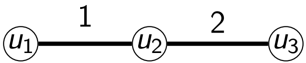
$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization
Input: A graph $G = (V, E)$, a duration matrix D, and a period Δ.
Question: Is there a temporal labeling $\lambda \colon E \to [1, \Delta]$ such that the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5$



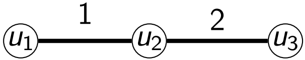
$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Periodic Temporal Graph Realization

We study the problem of realizing a periodic temporal graph introduced by Klobas et al. (SAND '24).

Periodic Temporal Graph Realization
Input: A graph $G = (V, E)$, a duration matrix D, and a period Δ.
Question: Is there a temporal labeling $\lambda\colon E \to [1, \Delta]$ such that the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5$



$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Multi-Label Periodic Temporal Graph Realization

Multi-Label Periodic TGR

Input: A graph $G = (V, E)$, a duration matrix D, a period Δ,

Multi-Label Periodic Temporal Graph Realization

Multi-Label Periodic TGR
Input: A graph $G = (V, E)$, a duration matrix D, a period Δ, and $\ell \in \mathbb{N}$.
Question:

Multi-Label Periodic Temporal Graph Realization

Multi-Label Periodic TGR
Input: A graph $G = (V, E)$, a duration matrix D, a period Δ, and $\ell \in \mathbb{N}$.
Question: Is there a temporal multi-labeling $\lambda \colon E \to 2^{[1,\Delta]}$ such that $1 \leq |\lambda(e)| \leq \ell$ for each edge $e \in E$ and the Δ-periodic graph (G, λ) realizes D?

Multi-Label Periodic TGR
Input: A graph $G = (V, E)$, a duration matrix D, a period Δ, and $\ell \in \mathbb{N}$.
Question: Is there a temporal multi-labeling $\lambda \colon E \to 2^{[1,\Delta]}$ such that $1 \leq |\lambda(e)| \leq \ell$ for each edge $e \in E$ and the Δ-periodic graph (G, λ) realizes D?

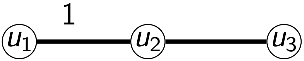
$\Delta = 5, \ell = 2$

$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Multi-Label Periodic Temporal Graph Realization

Multi-Label Periodic TGR
Input: A graph $G = (V, E)$, a duration matrix D, a period Δ, and $\ell \in \mathbb{N}$.
Question: Is there a temporal multi-labeling $\lambda \colon E \to 2^{[1,\Delta]}$ such that $1 \leq |\lambda(e)| \leq \ell$ for each edge $e \in E$ and the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5, \ell = 2$

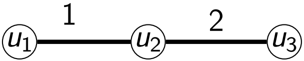


$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Multi-Label Periodic Temporal Graph Realization

Multi-Label Periodic TGR
Input: A graph $G = (V, E)$, a duration matrix D, a period Δ, and $\ell \in \mathbb{N}$.
Question: Is there a temporal multi-labeling $\lambda\colon E \to 2^{[1,\Delta]}$ such that $1 \le |\lambda(e)| \le \ell$ for each edge $e \in E$ and the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5, \ell = 2$



$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

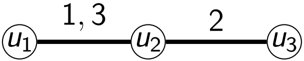
Multi-Label Periodic Temporal Graph Realization

Multi-Label Periodic TGR

Input: A graph $G = (V, E)$, a duration matrix D, a period Δ, and $\ell \in \mathbb{N}$.

Question: Is there a temporal multi-labeling $\lambda \colon E \to 2^{[1,\Delta]}$ such that $1 \leq |\lambda(e)| \leq \ell$ for each edge $e \in E$ and the Δ-periodic graph (G, λ) realizes D?

$\Delta = 5, \ell = 2$



$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Hardness of Multi-Label Periodic TGR

Theorem (Klobas et al., SAND '24)

Periodic TGR is NP-hard for each $\Delta \geq 3$.

Hardness of Multi-Label Periodic TGR

Theorem (Klobas et al., SAND '24)

Periodic TGR is NP-hard for each $\Delta \geq 3$.

Theorem

For each $\ell \geq 1$, *Multi-Label Periodic TGR is NP-hard even on split graphs with* $\Delta = 6$ *and* $d_{\max} = 3$.

Hardness of Multi-Label Periodic TGR

Theorem (Klobas et al., SAND '24)

Periodic TGR is NP-hard for each $\Delta \geq 3$.

Theorem

For each $\ell \geq 1$, Multi-Label Periodic TGR is NP-hard even on split graphs with $\Delta = 6$ and $d_{\max} = 3$.

Theorem (Klobas et al., SAND '24)

Periodic TGR is polynomial-time solvable for trees and FPT with respect to feedback edge number.

Hardness of Multi-Label Periodic TGR

Theorem (Klobas et al., SAND '24)

Periodic TGR is NP-hard for each $\Delta \geq 3$.

Theorem

For each $\ell \geq 1$, *Multi-Label Periodic TGR is NP-hard even on split graphs with* $\Delta = 6$ *and* $d_{\max} = 3$.

Theorem (Klobas et al., SAND '24)

Periodic TGR is polynomial-time solvable for trees and FPT with respect to feedback edge number.

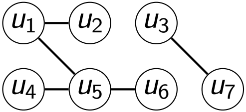
Theorem

For each $\ell \geq 5$, *Multi-Label Periodic TGR is NP-hard on stars.*

Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover** ("is there a **vertex cover** of size k in the graph G?")

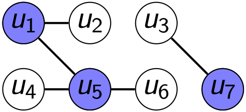
$k = 3$



Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover** ("is there a **vertex cover** of size k in the graph G?")

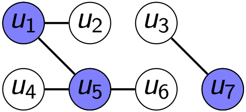
$k = 3$

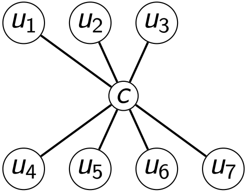


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover** ("is there a **vertex cover** of size k in the graph G?")

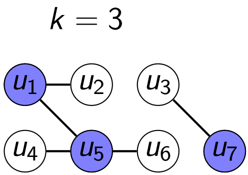
$k = 3$

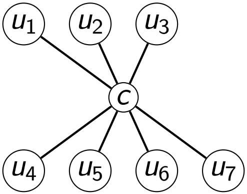




Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**





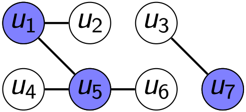
$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

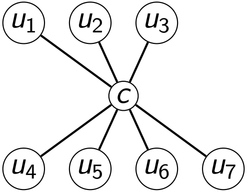
$$\Delta \approx (k + 2) \cdot n^2$$

Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

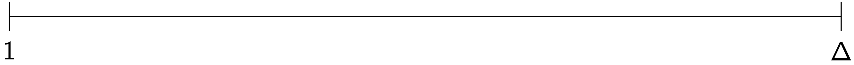
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

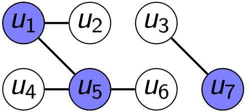
$$\Delta \approx (k + 2) \cdot n^2$$

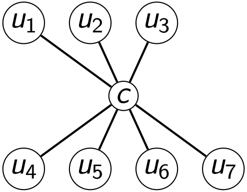


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

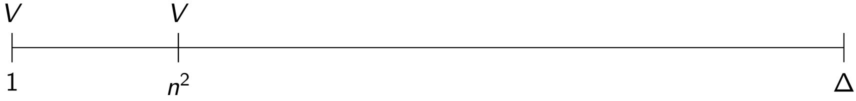
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

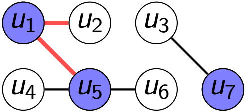
$$\Delta \approx (k + 2) \cdot n^2$$

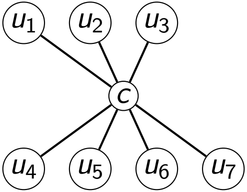


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

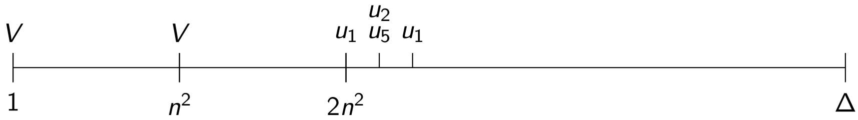
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

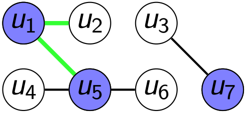
$$\Delta \approx (k+2) \cdot n^2$$

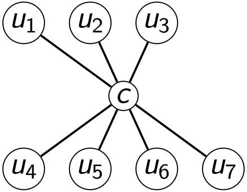


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

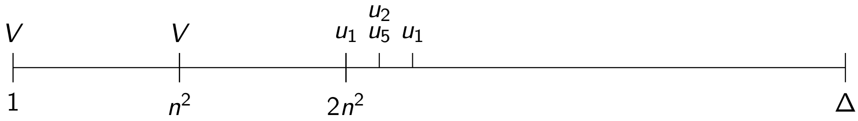
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

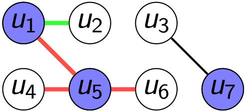
$$\Delta \approx (k+2) \cdot n^2$$

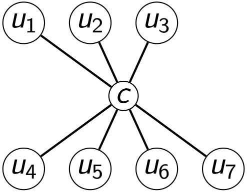


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

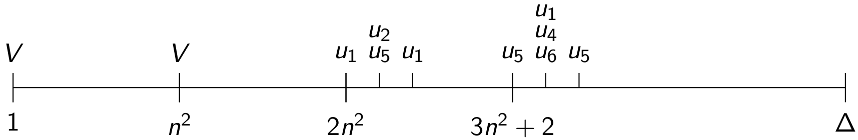
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

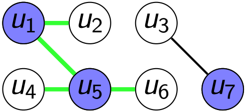
$$\Delta \approx (k+2) \cdot n^2$$

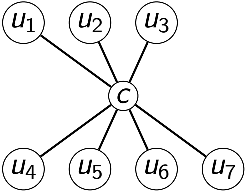


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

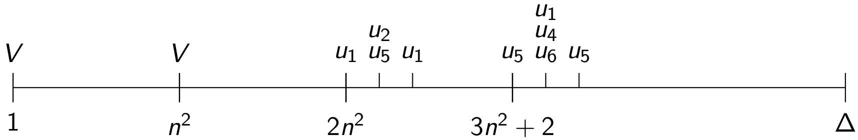
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

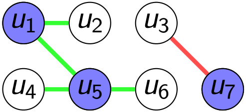
$$\Delta \approx (k + 2) \cdot n^2$$

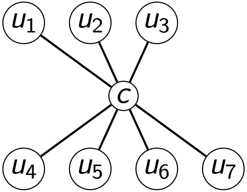


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

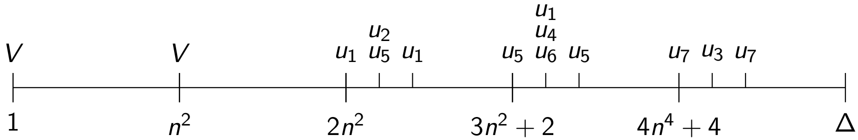
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

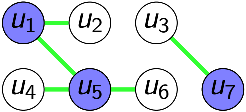
$$\Delta \approx (k+2) \cdot n^2$$

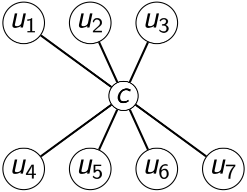


Ideas of the reduction for hardness on stars

Reduction from **Vertex Cover**

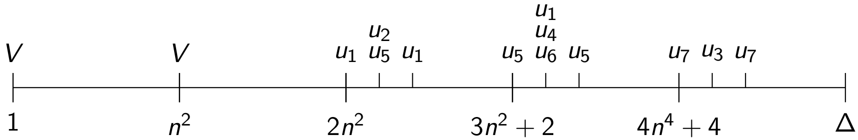
$k = 3$





$$D_{vw} = \begin{cases} 2 & \{v, w\} \in E \\ n^2 & \{v, w\} \notin E \end{cases}$$

$$\Delta \approx (k + 2) \cdot n^2$$



Algorithmic results

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

$\mathrm{vc} \mathrel{\hat{=}}$ vertex cover number

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

$\rightsquigarrow$ Consider larger parameter combinations!

$\mathrm{vc} \hat{=}$ vertex cover number

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

$\rightsquigarrow$ Consider larger parameter combinations!

Theorem

Multi-Label Periodic TGR is FPT for

$\mathrm{vc} \triangleq$ vertex cover number

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

$\rightsquigarrow$ Consider larger parameter combinations!

Theorem

Multi-Label Periodic TGR is FPT for

- *the number n of vertices and*

vc $\hat{=}$ vertex cover number

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

$\rightsquigarrow$ Consider larger parameter combinations!

Theorem

Multi-Label Periodic TGR is FPT for

- *the number n of vertices and*
- $\mathrm{vc} + \Delta$.

$\mathrm{vc} \mathrel{\hat{=}}$ vertex cover number

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

$\rightsquigarrow$ Consider larger parameter combinations!

Theorem

Multi-Label Periodic TGR is FPT for

- *the number n of vertices and*
- $\mathrm{vc} + \Delta$.

Theorem

Multi-Label Periodic TGR admits a polynomial kernel for $\mathrm{nu} + d_{\max}$.

$\mathrm{vc} \mathrel{\hat{=}}$ vertex cover number

Algorithmic results

Implications: No FPT-algorithms for (i) $k := \ell + \mathrm{vc}$ or (ii) $k := \ell + d_{\max}$.

$\rightsquigarrow$ Consider larger parameter combinations!

Theorem

Multi-Label Periodic TGR is FPT for

- *the number n of vertices and*
- $\mathrm{vc} + \Delta$.

Theorem

Multi-Label Periodic TGR admits a polynomial kernel for $\mathrm{nu} + d_{\max}$.

$\mathrm{vc} \mathrel{\hat{=}}$ vertex cover number, $\mathrm{nu} \mathrel{\hat{=}}$ number of non-universal vertices

An FPT-algorithm for n

An FPT-algorithm for n

W.l.o.g.: $\ell \leq n^2$

An FPT-algorithm for n

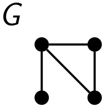
W.l.o.g.: $\ell \leq n^2$ (if an edge receives $> n^2$ labels, one can be omitted)

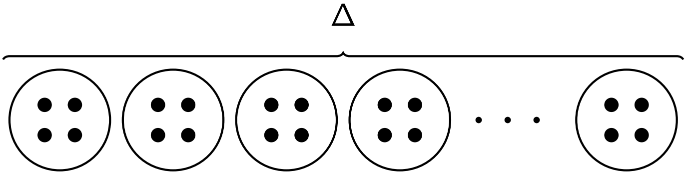
An FPT-algorithm for n

W.l.o.g.: $\ell \leq n^2$ (if an edge receives $> n^2$ labels, one can be omitted)
$\rightsquigarrow$ If there is a solution, then there is one with at most $n^2 \cdot \ell \leq n^4$ non-empty snapshots

An FPT-algorithm for n

W.l.o.g.: $\ell \leq n^2$ (if an edge receives $> n^2$ labels, one can be omitted)
$\rightsquigarrow$ If there is a solution, then there is one with at most $n^2 \cdot \ell \leq n^4$ non-empty snapshots

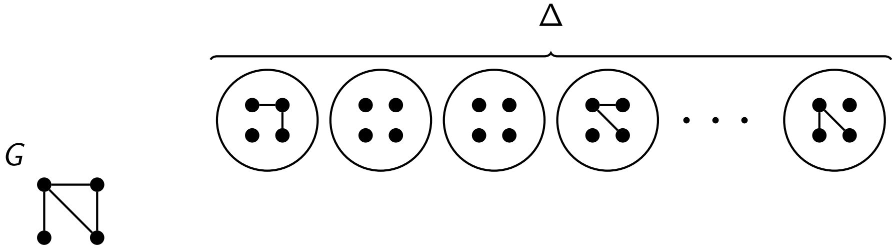




An FPT-algorithm for n

W.l.o.g.: $\ell \leq n^2$ (if an edge receives $> n^2$ labels, one can be omitted)

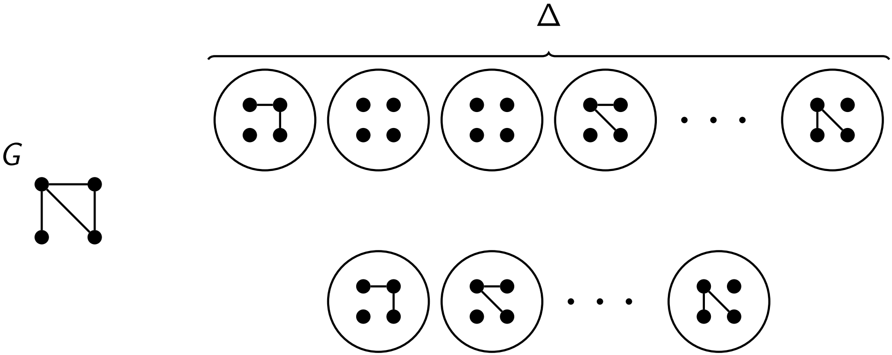
$\rightsquigarrow$ If there is a solution, then there is one with at most $n^2 \cdot \ell \leq n^4$ non-empty snapshots



An FPT-algorithm for n

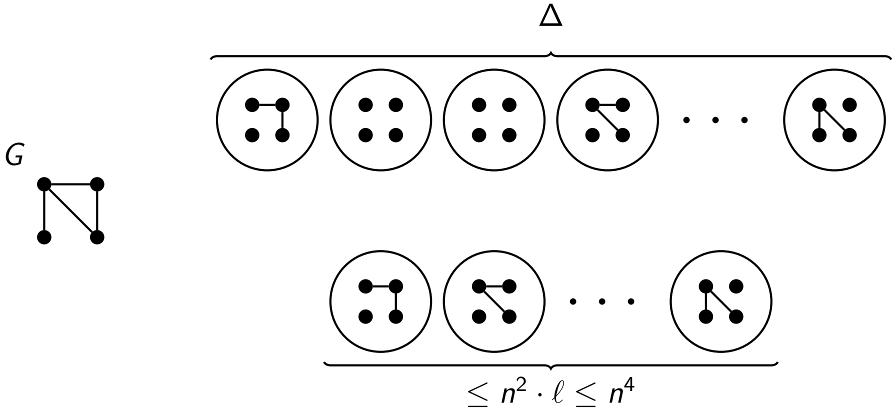
W.l.o.g.: $\ell \leq n^2$ (if an edge receives $> n^2$ labels, one can be omitted)

$\rightsquigarrow$ If there is a solution, then there is one with at most $n^2 \cdot \ell \leq n^4$ non-empty snapshots



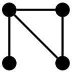
An FPT-algorithm for n

W.l.o.g.: $\ell \leq n^2$ (if an edge receives $> n^2$ labels, one can be omitted)
$\rightsquigarrow$ If there is a solution, then there is one with at most $n^2 \cdot \ell \leq n^4$ non-empty snapshots

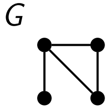


An FPT-algorithm for n

G

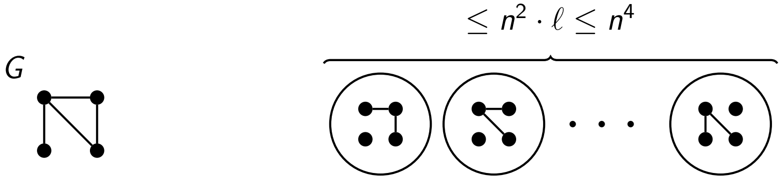


An FPT-algorithm for n



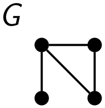
Guess the $\mathcal{O}(n^4)$ non-empty snapshots

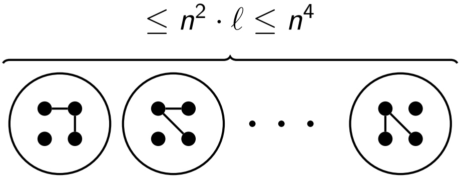
An FPT-algorithm for n



Guess the $\mathcal{O}(n^4)$ non-empty snapshots (in $n^{\mathcal{O}(n^4)}$ time)

An FPT-algorithm for n

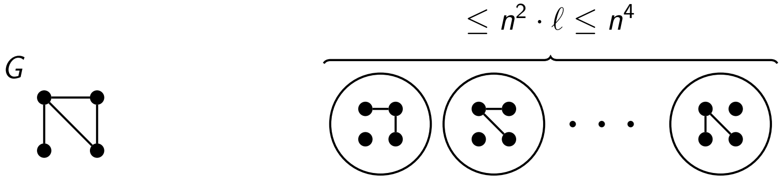




Guess the $\mathcal{O}(n^4)$ non-empty snapshots (in $n^{\mathcal{O}(n^4)}$ time)

Remaining task:

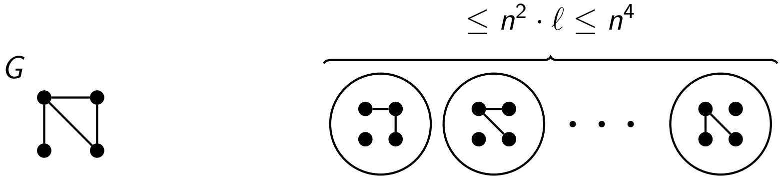
An FPT-algorithm for n



Guess the $\mathcal{O}(n^4)$ non-empty snapshots (in $n^{\mathcal{O}(n^4)}$ time)

Remaining task: Determine number of empty snapshots between any two non-empty ones

An FPT-algorithm for n

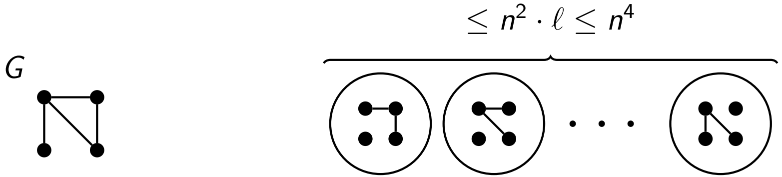


Guess the $\mathcal{O}(n^4)$ non-empty snapshots (in $n^{\mathcal{O}(n^4)}$ time)

Remaining task: Determine number of empty snapshots between any two non-empty ones

$\rightsquigarrow$ Use an ILP with $\mathcal{O}(n^4)$ variables

An FPT-algorithm for n

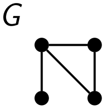


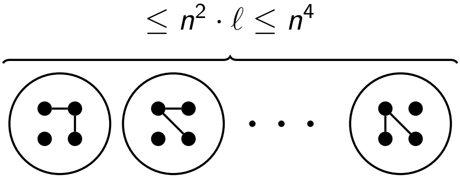
Guess the $\mathcal{O}(n^4)$ non-empty snapshots (in $n^{\mathcal{O}(n^4)}$ time)

Remaining task: Determine number of empty snapshots between any two non-empty ones

$\rightsquigarrow$ Use an ILP with $\mathcal{O}(n^4)$ variables (takes $n^{\mathcal{O}(n^4)}$ time)

An FPT-algorithm for n





Guess the $\mathcal{O}(n^4)$ non-empty snapshots (in $n^{\mathcal{O}(n^4)}$ time)

Remaining task: Determine number of empty snapshots between any two non-empty ones

$\rightsquigarrow$ Use an ILP with $\mathcal{O}(n^4)$ variables (takes $n^{\mathcal{O}(n^4)}$ time)

Theorem

Multi-Label Periodic TGR can be solved in $n^{\mathcal{O}(n^4)}$ time.

Details of the ILP

Determine whether a set L of non-empty snapshots can be assigned to time steps so that the temporal graph realizes D:

- Variables t_i: time step for i-th snapshot

Details of the ILP

Determine whether a set L of non-empty snapshots can be assigned to time steps so that the temporal graph realizes D:

- Variables t_i: time step for i-th snapshot
- Guess s_{uv} for all (u, v): The layer in which the journey realizing D_{uv} starts.

Details of the ILP

Determine whether a set L of non-empty snapshots can be assigned to time steps so that the temporal graph realizes D:

- Variables t_i: time step for i-th snapshot
- Guess s_{uv} for all (u, v): The layer in which the journey realizing D_{uv} starts.
- Calculate $a(u, v, i) = (j, z)$ so that fastest u-v-journey starting in i-th layer reaches v in j-th layer z periods later

Details of the ILP

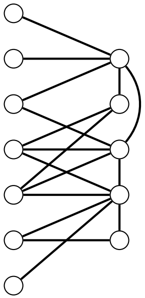
Determine whether a set L of non-empty snapshots can be assigned to time steps so that the temporal graph realizes D:

- Variables t_i: time step for i-th snapshot
- Guess s_{uv} for all (u, v): The layer in which the journey realizing D_{uv} starts.
- Calculate $a(u, v, i) = (j, z)$ so that fastest u-v-journey starting in i-th layer reaches v in j-th layer z periods later

ILP:

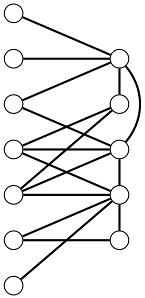
$$
\begin{aligned}
t_j + z\Delta - t_i + 1 &= D_{uv}, && \forall (u, v), i = s_{uv}, a(u, v, i) = (j, z) \\
t_j + z\Delta - t_i + 1 &\geq D_{uv}, && \forall (u, v), i \neq s_{uv}, a(u, v, i) = (j, z) \\
t_1 &\geq 1 \\
t_i - t_{i-1} &\geq 1, && \forall i : 2 \leq i \leq L \\
t_L &\leq \Delta
\end{aligned}
$$

Parameter combination vc + Δ



Parameter combination vc + Δ

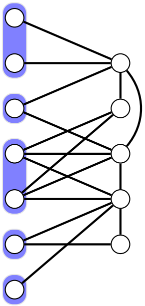
Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$



Parameter combination vc + Δ

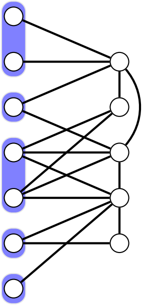
Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

The number of **neighborhood classes** is bounded



Parameter combination vc + Δ

Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$



The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

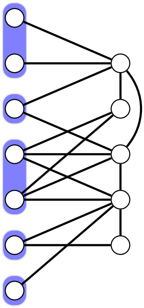
Parameter combination vc + Δ

Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

C contains **label twins** u and v



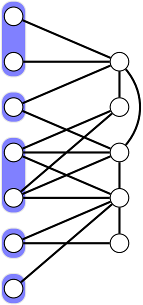
Parameter combination vc + Δ

Goal: Reduce the number of vertices to at most $f(\mathrm{vc}+\Delta)$

The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

C contains **label twins** u and v (i.e. $\lambda(\{u,w\}) = \lambda(\{v,w\})$ for each $w \in N(v)$)



Parameter combination vc + Δ

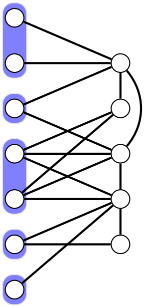
Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

C contains **label twins** u and v (i.e. $\lambda(\{u, w\}) = \lambda(\{v, w\})$ for each $w \in N(v)$)

The rows (columns) of label twins in D are identical



Parameter combination vc + Δ

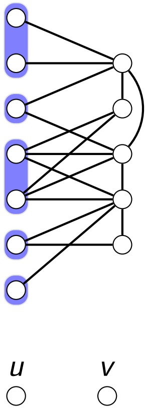
Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

The number of **neighborhood classes** is bounded

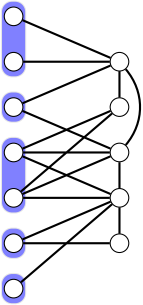
How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

C contains **label twins** u and v (i.e. $\lambda(\{u, w\}) = \lambda(\{v, w\})$ for each $w \in N(v)$)

The rows (columns) of label twins in D are identical



Parameter combination vc + Δ



Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

The number of **neighborhood classes** is bounded

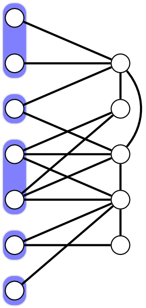
How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

C contains **label twins** u and v (i.e. $\lambda(\{u, w\}) = \lambda(\{v, w\})$ for each $w \in N(v)$)

The rows (columns) of label twins in D are identical

Parameter combination vc + Δ

Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

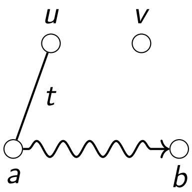


The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

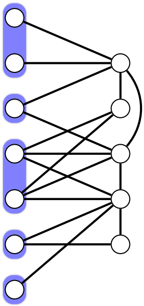
C contains **label twins** u and v (i.e. $\lambda(\{u, w\}) = \lambda(\{v, w\})$ for each $w \in N(v)$)

The rows (columns) of label twins in D are identical



Parameter combination vc + Δ

Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

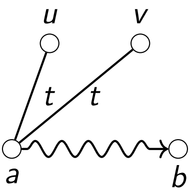


The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

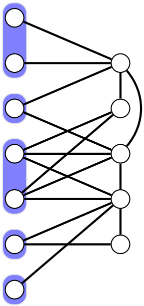
C contains **label twins** u and v (i.e. $\lambda(\{u, w\}) = \lambda(\{v, w\})$ for each $w \in N(v)$)

The rows (columns) of label twins in D are identical



Parameter combination vc + Δ

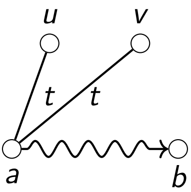
Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$



The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

C contains **label twins** u and v (i.e. $\lambda(\{u,w\}) = \lambda(\{v,w\})$ for each $w \in N(v)$)



The rows (columns) of label twins in D are identical

Detect identical rows & columns and remove one

Parameter combination vc + Δ

Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$

The number of **neighborhood classes** is bounded

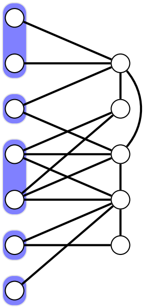
How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

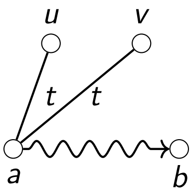
C contains **label twins** u and v (i.e. $\lambda(\{u, w\}) = \lambda(\{v, w\})$ for each $w \in N(v)$)

The rows (columns) of label twins in D are identical

Detect identical rows & columns and remove one

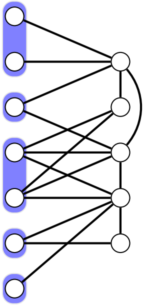
$\rightsquigarrow$ vertex-kernel of size $2^{\mathcal{O}(\Delta \cdot \mathrm{vc})}$





Parameter combination vc + Δ

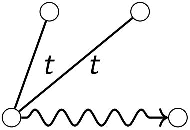
Goal: Reduce the number of vertices to at most $f(\mathrm{vc} + \Delta)$



The number of **neighborhood classes** is bounded

How to reduce the size of a neighborhood class C with $|C| > 2^{\Delta \cdot \mathrm{vc}}$?

C contains **label twins** u and v (i.e. $\lambda(\{u, w\}) = \lambda(\{v, w\})$ for each $w \in N(v)$)



The rows (columns) of label twins in D are identical

Detect identical rows & columns and remove one

$\rightsquigarrow$ vertex-kernel of size $2^{\mathcal{O}(\Delta \cdot \mathrm{vc})}$ (or $\Delta^{\mathcal{O}(\ell \cdot \mathrm{vc})}$)

Conclusion and outlook

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Multi-Label Periodic TGR is NP-hard on stars for $\ell \geq 5$ and in P on general trees for $\ell = 1$:

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Multi-Label Periodic TGR is NP-hard on stars for $\ell \geq 5$ and in P on general trees for $\ell = 1$: What is the complexity for $\ell \in \{2, 3, 4\}$?

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Multi-Label Periodic TGR is NP-hard on stars for $\ell \geq 5$ and in P on general trees for $\ell = 1$: What is the complexity for $\ell \in \{2, 3, 4\}$?

Find other parameters for which the problem is FPT.

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Multi-Label Periodic TGR is NP-hard on stars for $\ell \geq 5$ and in P on general trees for $\ell = 1$: What is the complexity for $\ell \in \{2, 3, 4\}$?

Find other parameters for which the problem is FPT.

Is there a polynomial kernel for *n*?

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Multi-Label Periodic TGR is NP-hard on stars for $\ell \geq 5$ and in P on general trees for $\ell = 1$: What is the complexity for $\ell \in \{2, 3, 4\}$?

Find other parameters for which the problem is FPT.

Is there a polynomial kernel for n?

Consider other variations of the problem:

- D specifies only an upper bound and not the exact duration.

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Multi-Label Periodic TGR is NP-hard on stars for $\ell \geq 5$ and in P on general trees for $\ell = 1$: What is the complexity for $\ell \in \{2, 3, 4\}$?

Find other parameters for which the problem is FPT.

Is there a polynomial kernel for n?

Consider other variations of the problem:

- D specifies only an upper bound and not the exact duration.
- Entries of D may contain more than one value, the duration of the fastest paths must be one of these values.

Conclusion and outlook

We developed parameterized algorithms for Multi-Label Periodic TGR (and a non-periodic version).

Multi-Label Periodic TGR is NP-hard on stars for $\ell \geq 5$ and in P on general trees for $\ell = 1$: What is the complexity for $\ell \in \{2, 3, 4\}$?

Find other parameters for which the problem is FPT.

Is there a polynomial kernel for n?

Consider other variations of the problem:

- D specifies only an upper bound and not the exact duration.
- Entries of D may contain more than one value, the duration of the fastest paths must be one of these values.
- Consider non-strict instead of strict temporal paths.

Thank you!

Questions?