# Twin-width: Forbidden subdivisions & Polynomial kernels

Amadeus Reinald
ENS de Lyon

*joint work with*
Édouard Bonnet, Eun Jung Kim,
Stéphan Thomassé, Rémi Watrigant

Séminaire ALGCO
December 2nd, 2021, Montpellier

# Graph parameters

Two definitions of **queue-number**:

# Graph parameters

Two definitions of **queue-number**:

**Structural:** vertex **ordering** + edge colouring / no nesting:

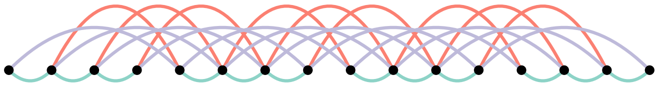


Fig. 2-queue layout of a triangulated grid [Duj+21].

# Graph parameters

Two definitions of **queue-number**:

**Structural:** vertex **ordering** + edge colouring / no nesting:

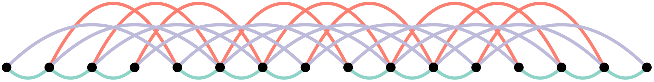
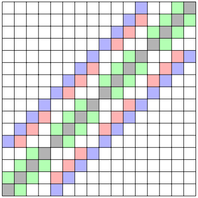


Fig. 2-queue layout of a triangulated grid [Duj+21].

**Matricial:** Adjacency matrix coloured by $k$ increasing zones



# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.

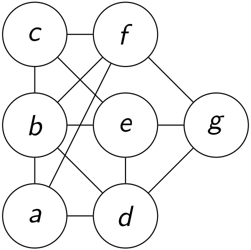



Fig. A (2)-contraction-sequence

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.

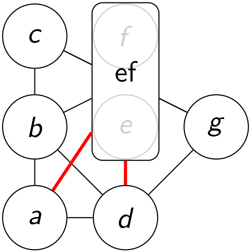



Fig. A (2)-contraction-sequence

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.

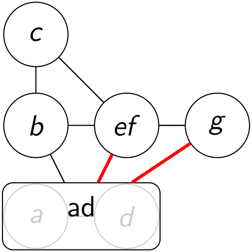



Fig. A (2)-contraction-sequence

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.

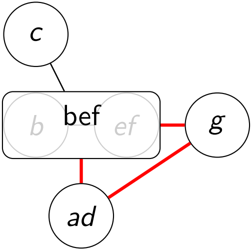



Fig. A (2)-contraction-sequence

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.

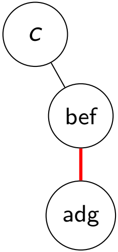



Fig. A (2)-contraction-sequence

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.

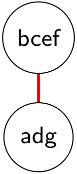



Fig. A (2)-contraction-sequence

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.


Fig. A (2)-contraction-sequence

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.


Fig. A (2)-contraction-sequence

$d$-contraction sequence : maximal red degree $d$.

# Twin-width: contraction sequences

Structural def: contract near-twins, record neighbourhood errors.


Fig. A (2)-contraction-sequence

$d$-contraction sequence : maximal red degree $d$.

**Definition** *The twin-width of $G$ is the least integer $d$ such that $G$ admits a $d$-contraction sequence.*

# Twin-width: mixed minors

Matrix def:

# Twin-width: mixed minors

Matrix def:

**Theorem (Bonnet, Kim, Thomassé, Watrigant '20)**
*A class of graphs has bounded tww if and only if it is $t$-mixed free for some constant $t$.*

$$\left[\begin{array}{cc|ccc|ccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

Fig. A 3 mixed-minor and a 4-grid minor

# Twin-width: mixed minors

Matrix def:

> **Theorem (Bonnet, Kim, Thomassé, Watrigant ’20)**
> *A class of graphs has bounded tww if and only if it is $t$-mixed free for some constant $t$.*

$$\left[\begin{array}{cc|ccc|ccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]
\qquad
\left[\begin{array}{cc|cc|cc|cc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

Fig. A 3 mixed-minor and a 4-grid minor

# Twin-width: Implications

How vast are bounded twin-width classes ?

# Twin-width: Implications

How vast are bounded twin-width classes ?

- Bounded treewidth, rank-width, queue/stack number ⇒ Bounded tww.

# Twin-width: Implications

How vast are bounded twin-width classes ?

- Bounded treewidth, rank-width, queue/stack number ⇒ Bounded tww.
- Proper minor-closed classes (⊇ Grids), map graphs...

# Twin-width: Implications

How vast are bounded twin-width classes ?

- Bounded treewidth, rank-width, queue/stack number $\Rightarrow$ Bounded tww.
- Proper minor-closed classes ($\supseteq$ Grids), map graphs...

Algorithmic implications:

**Theorem (Bonnet, Kim, Thomassé, Watrigant '20)**
*For a graph $G$, given with a $d$-contraction sequence, and any FO formula $\phi$, deciding $G \models \phi$ can be done in (FPT) time $f(|\phi|, d) \cdot n$.*

# Twin-width: Implications

How vast are bounded twin-width classes ?

- Bounded treewidth, rank-width, queue/stack number ⇒ Bounded tww.
- Proper minor-closed classes (⊇ Grids), map graphs...

Algorithmic implications:

> **Theorem (Bonnet, Kim, Thomassé, Watrigant '20)**
> *For a graph $G$, given with a $d$-contraction sequence, and any FO formula $\phi$, deciding $G \models \phi$ can be done in (FPT) time $f(|\phi|, d) \cdot n$.*

$\phi$ can capture: Dominating Set, Vertex Cover, Independent Set...

# Motivations

Bounded tww: FO problems are decidable in FPT time.

# Motivations

Bounded tww: FO problems are decidable in FPT time.

1. Structural characterizations of bounded tww classes?

# Motivations

Bounded tww: FO problems are decidable in FPT time.

1. Structural characterizations of bounded tww classes?
2. Better complexity: existence of polynomial kernels?

# In this talk

Structure:

**Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*Theta-free graphs of girth at least five have bounded tww.*

# In this talk

Structure:

> **Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
> *Theta-free graphs of girth at least five have bounded tww.*

Complexity:

> **Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
> *On graphs of twin-width at most four, $k$-*DOMINATING SET *does not admit polynomial kernels unless* coNP $\subseteq$ NP/*poly.*

# In this talk

Structure:

**Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*Theta-free graphs of girth at least five have bounded tww.*

Complexity:

**Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*On graphs of twin-width at most four, $k$-DOMINATING SET does not admit polynomial kernels unless* coNP ⊆ NP/*poly.*

**Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*On bounded tww, Connected k-VC admits a $O(k^{\frac{3}{2}})$ kernel.*

# Parameters and graph minors

Question: High parameter ⇒ forced structure/subgraph?

# Parameters and graph minors

Question: High parameter ⇒ forced structure/subgraph?
Forbidden subgraph ⇒ low parameter?

# Parameters and graph minors

Question: High parameter ⇒ forced structure/subgraph?
Forbidden subgraph ⇒ low parameter?

Known approach: forbid subgraphs as **minors**.

# Parameters and graph minors

Question: High parameter ⇒ forced structure/subgraph?
Forbidden subgraph ⇒ low parameter?

Known approach: forbid subgraphs as **minors**.

**Theorem (Treewidth: R,S '86)**
*Graphs forbidding a wall* ***minor*** *have bounded treewidth.*

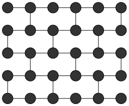


Fig. A wall

# Parameters and graph minors

Question: High parameter ⇒ forced structure/subgraph?
Forbidden subgraph ⇒ low parameter?

Known approach: forbid subgraphs as **minors**.

**Theorem (Treewidth: R,S '86)**
*Graphs forbidding a wall **minor** have bounded treewidth.*

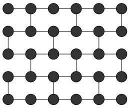


Fig. A wall

**Theorem (Twin-width: B,K,T,W '20)**
*Proper minor-closed classes have bounded twin-width.*

# Treewidth and induced subgraphs

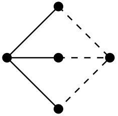
Generalize: forbid **induced** graph & its **subdivisions**.

# Treewidth and induced subgraphs

Generalize: forbid **induced** graph & its **subdivisions**.

Sparse + forbidden subwall ⇒ bounded treewidth?

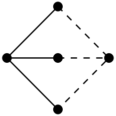
# Treewidth and induced subgraphs



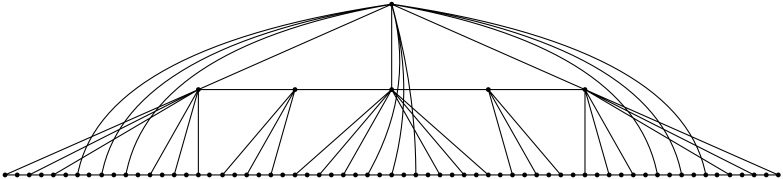
Generalize: forbid **induced** graph & its **subdivisions**.
Sparse + forbidden subwall ⇒ bounded treewidth?

# Treewidth and induced subgraphs

Generalize: forbid **induced** graph & its **subdivisions**.
Sparse + forbidden subwall ⇒ bounded treewidth?



**Theorem (Sintiari, Trotignon '19)**
*There exist graphs forbidding induced thetas, having arbitrarily large girth and treewidth.*



# Treewidth and induced subgraphs

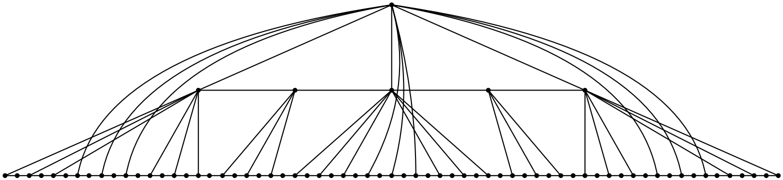


Fig. A layered wheel [ST19]

# Treewidth and induced subgraphs

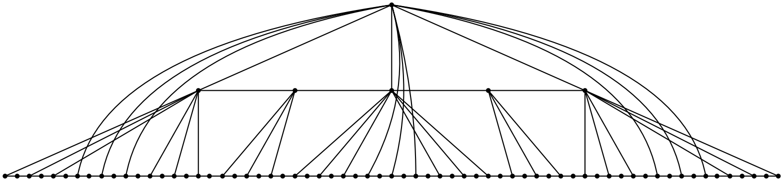


Fig. A layered wheel [ST19]

**Theorem (Abrishami, Chudnovsky, Hajebi, Spirkl '21)**
*There exists c s.t. for any (Theta, triangle)-free G,*
$tw(G) \leqslant c \log(|V(G)|)$.

# Treewidth and induced subgraphs

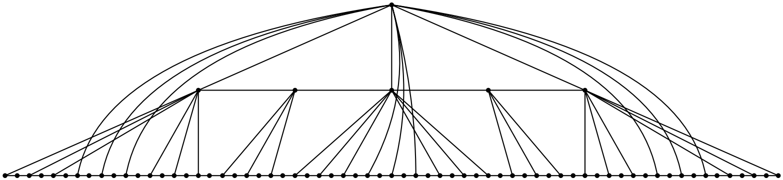


Fig. A layered wheel [ST19]

> **Theorem (Abrishami, Chudnovsky, Hajebi, Spirkl '21)**
> *There exists c s.t. for any (Theta, triangle)-free G,*
> $tw(G) \leqslant c \log(|V(G)|)$.

Do such graphs have bounded twin-width?

# Twin-width and induced subgraphs

How does twin-width motivate the study of theta-free classes?

# Twin-width and induced subgraphs

How does twin-width motivate the study of theta-free classes?

- We can only forbid subcubic graphs: subcubic graphs already have unbounded tww.

# Twin-width and induced subgraphs

How does twin-width motivate the study of theta-free classes?

- We can only forbid subcubic graphs: subcubic graphs already have unbounded tww.
- $\Omega(\log(n))$-subdivisions of cliques have bounded tww, while $o(\log(n))$ subdivisions have unbounded tww.

# Twin-width and induced subgraphs

How does twin-width motivate the study of theta-free classes?

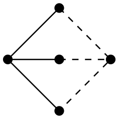
- We can only forbid subcubic graphs: subcubic graphs already have unbounded tww.
- $\Omega(\log(n))$-subdivisions of cliques have bounded tww, while $o(\log(n))$ subdivisions have unbounded tww.
- (excessively) Optimistic conjecture : forbiding any subcubic graph + sparse ⇒ bounded twin-width?

# Twin-width and induced subgraphs

How does twin-width motivate the study of theta-free classes?

- We can only forbid subcubic graphs: subcubic graphs already have unbounded tww.
- $\Omega(\log(n))$-subdivisions of cliques have bounded tww, while $o(\log(n))$ subdivisions have unbounded tww.
- (excessively) Optimistic conjecture : forbiding any subcubic graph + sparse ⇒ bounded twin-width?

Our class: Theta-free, girth 5.



# Roadmap

- Bounding tww ⇔ building a "simple", mixed-free **order**,

# Roadmap

- Bounding tww ⇔ building a "simple", mixed-free **order**,
- It suffices to rule out **grid** minors.

$$\left[\begin{array}{cc|cc|cc|cc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

# Roadmap

- Bounding tww ⇔ building a "simple", mixed-free **order**,
- It suffices to rule out **grid** minors.
- Build an order **witnessing enough structure** such that large grid minors ⇒ induced thetas.

$$\left[\begin{array}{cc|cc|cc|cc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

# Roadmap

- Bounding tww ⇔ building a "simple", mixed-free **order**,
- It suffices to rule out **grid** minors.
- Build an order **witnessing enough structure** such that large grid minors ⇒ induced thetas.

$$\left[\begin{array}{cc|cc|cc|cc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

Candidates: BFS, DFS...

# Roadmap

- Bounding tww ⇔ building a "simple", mixed-free **order**,
- It suffices to rule out **grid** minors.
- Build an order **witnessing enough structure** such that large grid minors ⇒ induced thetas.

$$\left[\begin{array}{cc|cc|cc|cc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

Candidates: BFS, DFS...
Little information on **connected** subgraphs, hard finding thetas.

# Connected BFS Decomposition

Idea: BFS exploring **connected parts** instead of vertices.

# Connected BFS Decomposition

Idea: BFS exploring **connected parts** instead of vertices.

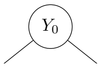
**Definition** *$Y$ is a* ***minimal connected neighbourhood*** *of $X$ if $N(X) \subseteq Y$, and $y, y' \in Y$ in the same CC of $G \setminus X$ are in the same CC of $G[Y]$.*

# Connected BFS Decomposition

Idea: BFS exploring **connected parts** instead of vertices.

> **Definition** *$Y$ is a **minimal connected neighbourhood** of $X$ if $N(X) \subseteq Y$, and $y, y' \in Y$ in the same CC of $G \backslash X$ are in the same CC of $G[Y]$.*

CBFS: $Y_0 = \{r\}$ for some $r \in V(G)$,

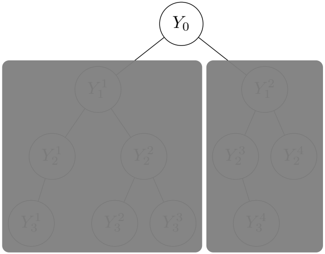


# Connected BFS Decomposition

Idea: BFS exploring **connected parts** instead of vertices.

**Definition** *$Y$ is a **minimal connected neighbourhood** of $X$ if $N(X) \subseteq Y$, and $y, y' \in Y$ in the same CC of $G \backslash X$ are in the same CC of $G[Y]$.*

CBFS: $Y_0 = \{r\}$ for some $r \in V(G)$, CBFS: $Y_{i+1} := mcn(Y_0 \cup \ldots \cup Y_i)$

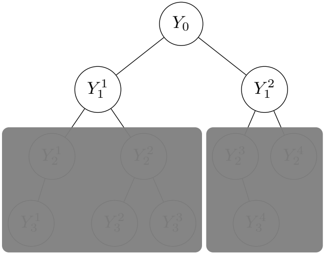


# Connected BFS Decomposition

Idea: BFS exploring **connected parts** instead of vertices.

**Definition** *$Y$ is a **minimal connected neighbourhood** of $X$ if $N(X) \subseteq Y$, and $y, y' \in Y$ in the same CC of $G \setminus X$ are in the same CC of $G[Y]$.*

CBFS: $Y_0 = \{r\}$ for some $r \in V(G)$, CBFS: $Y_{i+1} :=$ $mcn(Y_0 \cup \ldots \cup Y_i)$

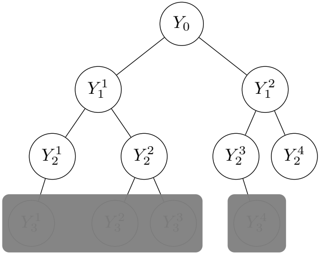


# Connected BFS Decomposition

Idea: BFS exploring **connected parts** instead of vertices.

**Definition** *$Y$ is a* ***minimal connected neighbourhood*** *of $X$ if $N(X) \subseteq Y$, and $y, y' \in Y$ in the same CC of $G \backslash X$ are in the same CC of $G[Y]$.*

CBFS: $Y_0 = \{r\}$ for some $r \in V(G)$, CBFS: $Y_{i+1} := mcn(Y_0 \cup \ldots \cup Y_i)$

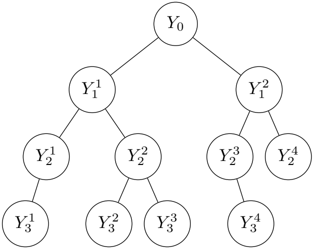


# Connected BFS Decomposition

Idea: BFS exploring **connected parts** instead of vertices.

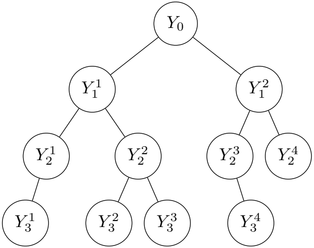
**Definition** *$Y$ is a* ***minimal connected neighbourhood*** *of $X$ if $N(X) \subseteq Y$, and $y, y' \in Y$ in the same CC of $G \backslash X$ are in the same CC of $G[Y]$.*

CBFS: $Y_0 = \{r\}$ for some $r \in V(G)$, CBFS: $Y_{i+1} := mcn(Y_0 \cup \ldots \cup Y_i)$



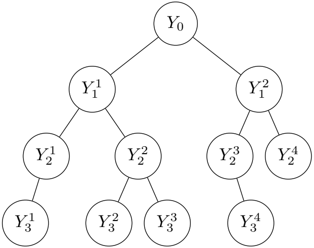
# Connected BFS Decomposition

$$Y_{i+1} := mcn(Y_0 \cup \ldots \cup Y_i)$$



# Connected BFS Decomposition

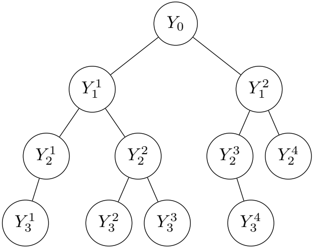
$$Y_{i+1} := mcn(Y_0 \cup \ldots \cup Y_i)$$




- Layer $Y_i$ adjacent only to $Y_{i-1}$ and $Y_{i+1}$,

# Connected BFS Decomposition

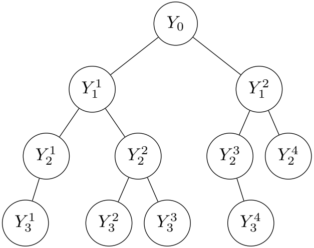
$$Y_{i+1} := mcn(Y_0 \cup \ldots \cup Y_i)$$




- Layer $Y_i$ adjacent only to $Y_{i-1}$ and $Y_{i+1}$,
- Component $Y_i^j$ has exactly **one antecedent** in $Y_{i-1}$.

# Connected BFS Decomposition

$$Y_{i+1} := mcn(Y_0 \cup \ldots \cup Y_i)$$




- Layer $Y_i$ adjacent only to $Y_{i-1}$ and $Y_{i+1}$,
- Component $Y_i^j$ has exactly **one antecedent** in $Y_{i-1}$.

Global structure: **tree** relating components $Y_i^j$.

# Chasing Complex Structures

Global relation between components is a tree, "simple" / tww.

# Chasing Complex Structures

Global relation between components is a tree, "simple" / tww.

Does complexity lie **inside** components $Y_i^j$?

# Chasing Complex Structures

Global relation between components is a tree, "simple" / tww.

Does complexity lie **inside** components $Y_i^j$? **No**

**Lemma** *For any CBFS of $G$, each $Y_i^j$ is a tree.*

# Chasing Complex Structures

Global relation between components is a tree, "simple" / tww.

Does complexity lie **inside** components $Y_i^j$? **No**

> **Lemma** *For any CBFS of $G$, each $Y_i^j$ is a tree. Moreover, each $v \in Y_i^j$ has an unique antecedent $v^{-1} \in Y_{i-1}$.*

# Chasing Complex Structures

Global relation between components is a tree, "simple" / tww.

Does complexity lie **inside** components $Y_i^j$? **No**

> **Lemma** *For any CBFS of $G$, each $Y_i^j$ is a tree. Moreover, each $v \in Y_i^j$ has an unique antecedent $v^{-1} \in Y_{i-1}$.*

$\hookrightarrow$ Complexity must lie in adjacencies **between** components...

# Chasing Complex Structures

Global relation between components is a tree, "simple" / tww.

Does complexity lie **inside** components $Y_i^j$? **No**

> **Lemma** *For any CBFS of $G$, each $Y_i^j$ is a tree. Moreover, each $v \in Y_i^j$ has an unique antecedent $v^{-1} \in Y_{i-1}$.*

$\hookrightarrow$ Complexity must lie in adjacencies **between** components...

1. Describe the **structure** between layers,

## Chasing Complex Structures

Global relation between components is a tree, "simple" / tww.

Does complexity lie **inside** components $Y_i^j$? **No**

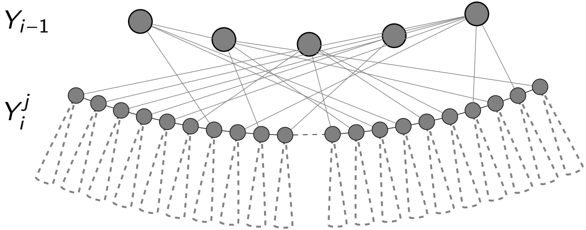
> **Lemma** *For any CBFS of $G$, each $Y_i^j$ is a tree. Moreover, each $v \in Y_i^j$ has an unique antecedent $v^{-1} \in Y_{i-1}$.*

↪ Complexity must lie in adjacencies **between** components...

1. Describe the **structure** between layers,
2. Use it to guide our **ordering** choice.

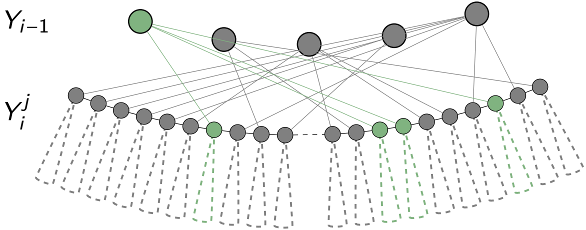
# Component-Antecedent Structure

How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?



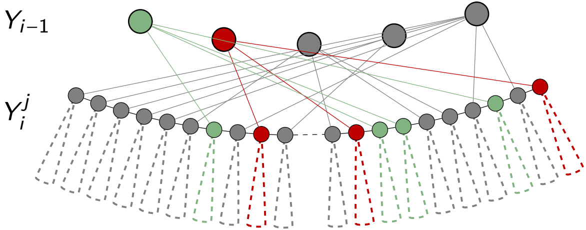
# Component-Antecedent Structure

How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?



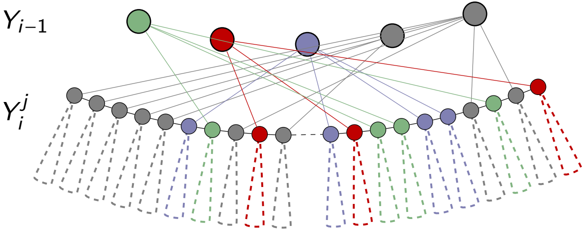
# Component-Antecedent Structure

How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?



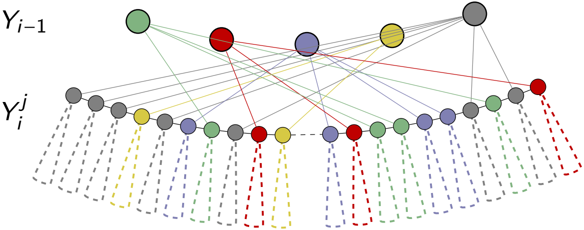
# Component-Antecedent Structure

How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?



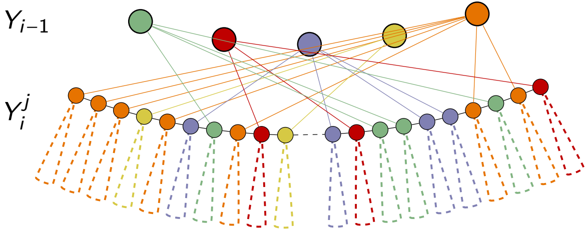
# Component-Antecedent Structure

How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?



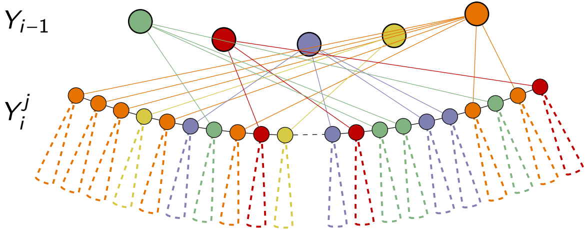
# Component-Antecedent Structure

How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?



# Component-Antecedent Structure

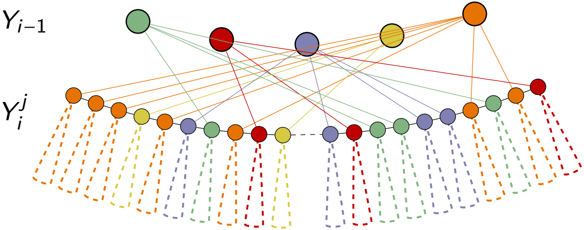
How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?




**Lemma** *For any $Y_i^j$, there is a* ***principal path*** *$P_i^j$ s.t. any successor $v \in Y_i^j$ belongs to a $v^{-1}$-private branch of $P_i^j$.*

# Component-Antecedent Structure

How do successors of **different** vertices of $Y_{i-1}$ relate in $Y_i$?




**Lemma** *For any $Y_i^j$, there is a **principal path** $P_i^j$ s.t. any successor $v \in Y_i^j$ belongs to a $v^{-1}$-private branch of $P_i^j$.*

Natural order: DFS exhausting private branches following $P_i^j$.

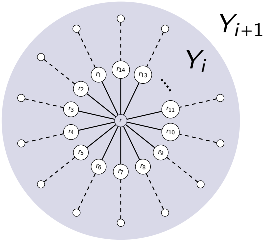
# Component-Successors Structure

**Locally:** for any vertex $r$ in component $Y_i^j$:

# Component-Successors Structure

**Locally:** for any vertex $r$ in component $Y_i^j$:

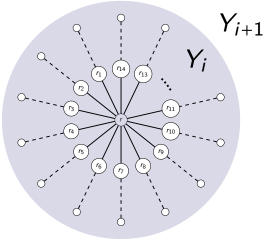
- $Y_i^j$ = tree → $r$-**branches are simple**,



# Component-Successors Structure

**Locally:** for any vertex $r$ in component $Y_i^j$:

- $Y_i^j$ = tree → $r$-**branches are simple**,
- Complexity = $r$-branches intertwining **using** $Y_{i+1}$,

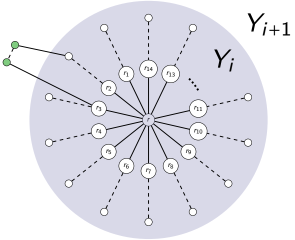


# Component-Successors Structure

**Locally:** for any vertex $r$ in component $Y_i^j$:

- $Y_i^j$ = tree → $r$-**branches are simple**,
- Complexity = $r$-branches intertwining **using** $Y_{i+1}$,

**Definition** *Consecutivity: shortest path between successors of different $r$-branches.*



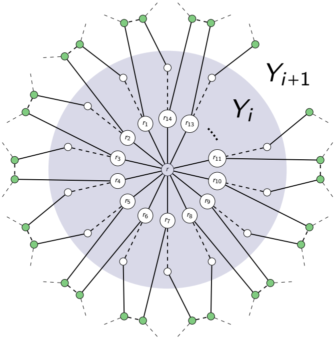
# Component-Successors Structure

**Locally:** for any vertex $r$ in component $Y_i^j$:

- $Y_i^j$ = tree → $r$-**branches are simple**,
- Complexity = $r$-branches intertwining **using** $Y_{i+1}$,

**Definition** *Consecutivity: shortest path between successors of different $r$-branches.*

**Lemma** *All but two $r$-branches admit at most two consecutivities.*



# Constructing the total order

- **Globally:** components $Y_i^j$ related as a tree

# Constructing the total order

- **Globally:** components $Y_i^j$ related as a tree
- → follow the **CBFS order** lexicographically.

# Constructing the total order

- **Globally:** components $Y_i^j$ related as a tree
- → follow the **CBFS order** lexicographically.

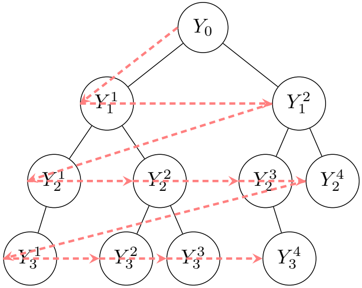



Fig. Inter-component **BFS**

# Constructing the total order

- **Globally:** components $Y_i^j$ related as a tree
- → follow the **CBFS order** lexicographically.

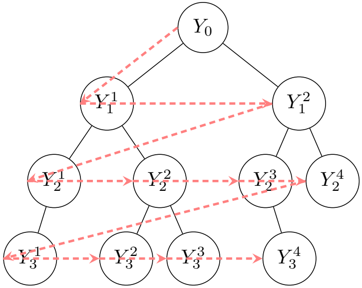



Fig. Inter-component **BFS**

- **Locally:** branches in $Y_i^j$: little intertwining on $Y_{i+1}$

# Constructing the total order

- **Globally:** components $Y_i^j$ related as a tree
- → follow the **CBFS order** lexicographically.

- **Locally:** branches in $Y_i^j$: little intertwining on $Y_{i+1}$
- → **DFS**: order branches / consec. cycles and paths.

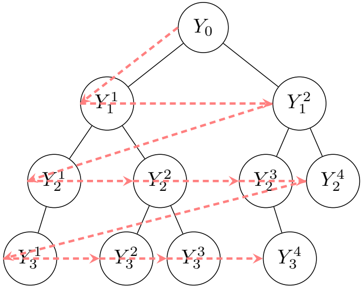



Fig. Inter-component **BFS**

# Constructing the total order

- **Globally:** components $Y_i^j$ related as a tree
- → follow the **CBFS order** lexicographically.

- **Locally:** branches in $Y_i^j$: little intertwining on $Y_{i+1}$
- → **DFS**: order branches / consec. cycles and paths.

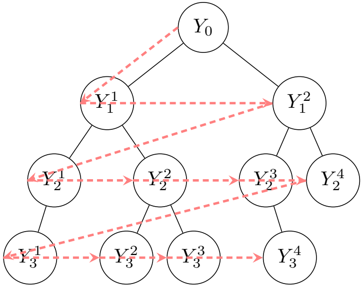



Fig. Inter-component **BFS**

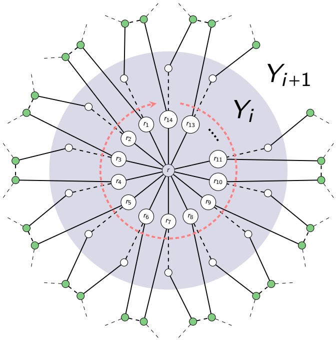



Fig. Intra-component DFS

# Bounding the twin-width

Graphs in the class → corresponding **ordered** matrices $M_<$:

# Bounding the twin-width

Graphs in the class → corresponding **ordered** matrices $M_<$:

Roadmap:

# Bounding the twin-width

Graphs in the class → corresponding **ordered** matrices $M_<$:

Roadmap:

1. Assume existence of arbitrarily large minors,

# Bounding the twin-width

Graphs in the class → corresponding **ordered** matrices $M_<$:

Roadmap:

1. Assume existence of arbitrarily large minors,
2. Use global order to localize them,

# Bounding the twin-width

Graphs in the class → corresponding **ordered** matrices $M_<$:

Roadmap:

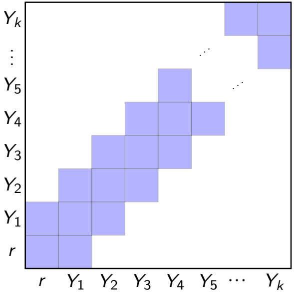
1. Assume existence of arbitrarily large minors,
2. Use global order to localize them,
3. Use local order to yield absurdity.

# Grid Minors Among Successive Layers

**Globally:** We have ordered layers $Y_0 < Y_1 < \dots < Y_k$, what does the adjacency matrix look like?

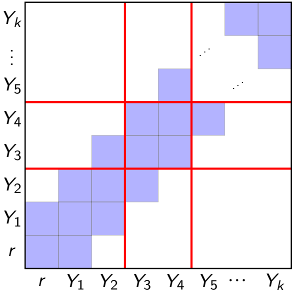
# Grid Minors Among Successive Layers

**Globally:** We have ordered layers $Y_0 < Y_1 < ... < Y_k$, what does the adjacency matrix look like?



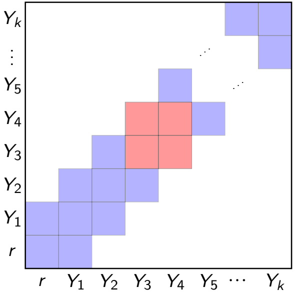
# Grid Minors Among Successive Layers

**Globally:** We have ordered layers $Y_0 < Y_1 < ... < Y_k$, what does the adjacency matrix look like?



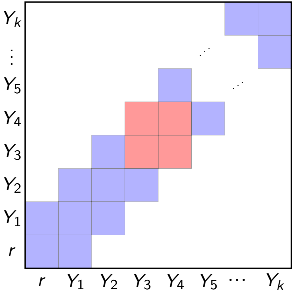
# Grid Minors Among Successive Layers

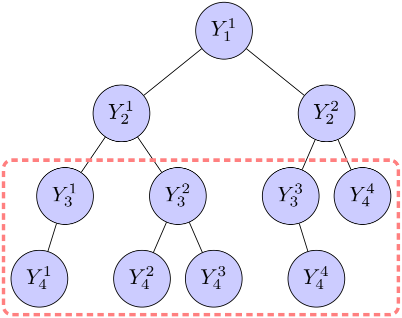
**Globally:** We have ordered layers $Y_0 < Y_1 < ... < Y_k$, what does the adjacency matrix look like?



# Grid Minors Among Successive Layers

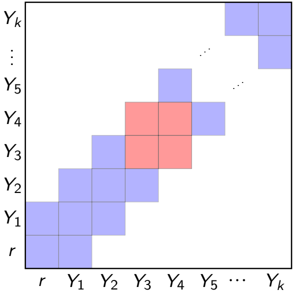
**Globally:** We have ordered layers $Y_0 < Y_1 < \ldots < Y_k$, what does the adjacency matrix look like?

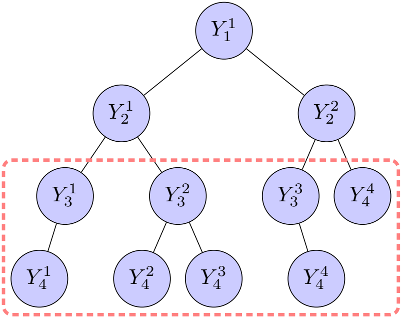


# Grid Minors Among Successive Layers

**Globally:** We have ordered layers $Y_0 < Y_1 < ... < Y_k$, what does the adjacency matrix look like?

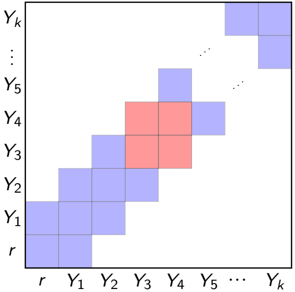



**Lemma** *If matrices $M_<$ have arbitrarily large grid minors, some submatrices indexed by $(Y_i \cup Y_{i+1})^2$ also do.*

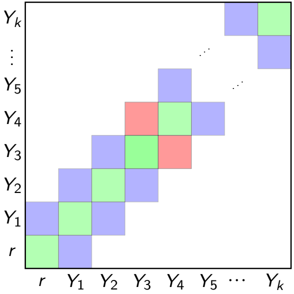
# Grid Minors Between Successive Layers

$Y_i$ are forests → bounded tww (DFS) →



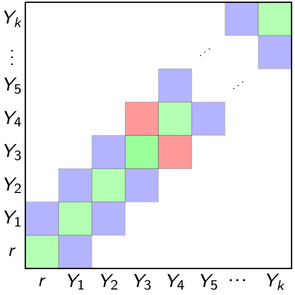
# Grid Minors Between Successive Layers

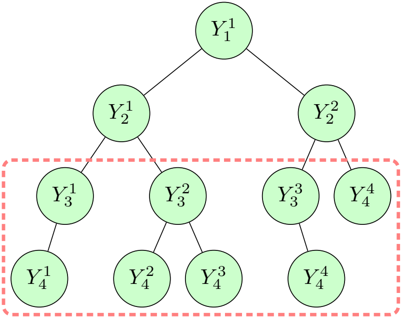
$Y_i$ are forests → bounded tww (DFS) → no large grid minors



# Grid Minors Between Successive Layers

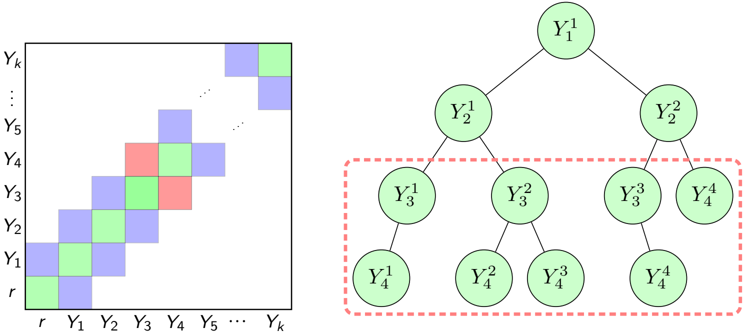
$Y_i$ are forests → bounded tww (DFS) → no large grid minors



# Grid Minors Between Successive Layers

$Y_i$ are forests → bounded tww (DFS) → no large grid minors



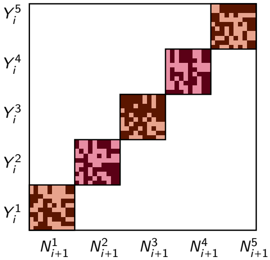

**Lemma** *If matrices $M_<$ have arbitrarily large grid minors, some submatrices indexed by $Y_i \times Y_{i+1}$ also do.*

# Grid Minors from One Component

Grid minor on $Y_i \times Y_{i+1}$ + lexicographical CBFS:

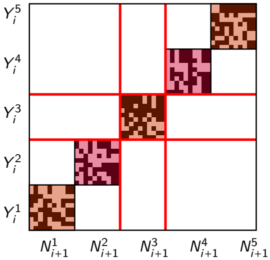
# Grid Minors from One Component

Grid minor on $Y_i \times Y_{i+1}$ + lexicographical CBFS:



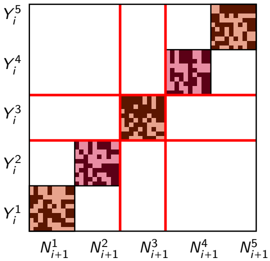
# Grid Minors from One Component

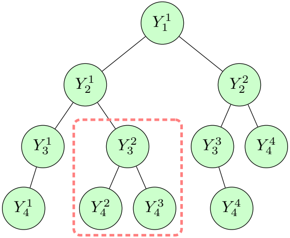
Grid minor on $Y_i \times Y_{i+1}$ + lexicographical CBFS:



# Grid Minors from One Component

Grid minor on $Y_i \times Y_{i+1}$ + lexicographical CBFS:




**Lemma** *If $M_<$ admit arbitrarily large grid minors, some submatrices indexed by $Y_i^j$ and its successors do too.*

# Excluding Grid Minors

Assume large grid minor on $Y_i^j$ and successors:

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | | $\vdots$ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |

# Excluding Grid Minors

Assume large grid minor on $Y_i^j$ and successors:

- Row sets $\leftrightarrow$ subforests of $Y_i^j$,

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | | $\vdots$ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |

# Excluding Grid Minors

Assume large grid minor on $Y_i^j$ and successors:

- Row sets ↔ subforests of $Y_i^j$,
- Column sets ↔ subpaths of $Y_{i+1}$,

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | | $\vdots$ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |

# Excluding Grid Minors

Assume large grid minor on $Y_i^j$ and successors:

- Row sets ↔ subforests of $Y_i^j$,
- Column sets ↔ subpaths of $Y_{i+1}$,
- Non-zero entries → **consecutivities**.

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | | $\vdots$ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |

# Excluding Grid Minors

Assume large grid minor on $Y_i^j$ and successors:

- Row sets ↔ subforests of $Y_i^j$,
- Column sets ↔ subpaths of $Y_{i+1}$,
- Non-zero entries → **consecutivities**.

Proof scheme:

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | | $\vdots$ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |

# Excluding Grid Minors

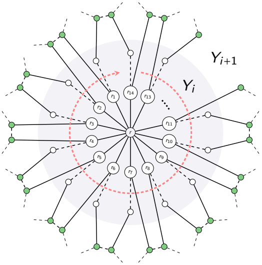
Assume large grid minor on $Y_i^j$ and successors:

- Row sets ↔ subforests of $Y_i^j$,
- Column sets ↔ subpaths of $Y_{i+1}$,
- Non-zero entries → **consecutivities**.

Proof scheme:

- Concentrate rows around $r \in Y_i^j$,

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| ⋮ | | | | ⋮ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | ⋯ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | ⋯ | $C_t$ |



# Excluding Grid Minors

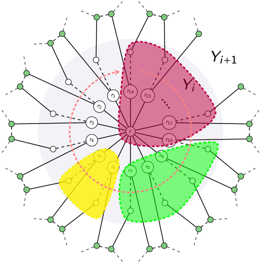
Assume large grid minor on $Y_i^j$ and successors:

- Row sets ↔ subforests of $Y_i^j$,
- Column sets ↔ subpaths of $Y_{i+1}$,
- Non-zero entries → **consecutivities**.

Proof scheme:

- Concentrate rows around $r \in Y_i^j$,

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | $\vdots$ | | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |



# Excluding Grid Minors

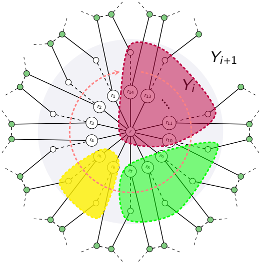
Assume large grid minor on $Y_i^j$ and successors:

- Row sets ↔ subforests of $Y_i^j$,
- Column sets ↔ subpaths of $Y_{i+1}$,
- Non-zero entries → **consecutivities**.

Proof scheme:

- Concentrate rows around $r \in Y_i^j$,
- Intertwining around $r$ **violates** our order

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | | $\vdots$ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |



# Excluding Grid Minors

Assume large grid minor on $Y_i^j$ and successors:

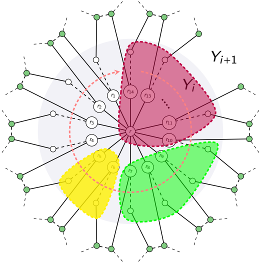
- Row sets ↔ subforests of $Y_i^j$,
- Column sets ↔ subpaths of $Y_{i+1}$,
- Non-zero entries → **consecutivities**.

Proof scheme:

- Concentrate rows around $r \in Y_i^j$,
- Intertwining around $r$ **violates** our order

**Theorem (B,K,R,T,W '21)**
*Theta-free graphs of girth at least 5 have bounded twin-width.*

| | | | | | | |
|---|---|---|---|---|---|---|
| $R_t$ | 1 | 1 | 1 | 1 | 1 | 1 |
| $\vdots$ | | | | $\vdots$ | | 1 |
| $R_3$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| $R_2$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| $R_1$ | 1 | 1 | 1 | 1 | | 1 |
| | $C_1$ | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_t$ |



# Further paths

- Do $\theta$-free, $g \geqslant 5$ graphs have **bounded queue/stack number**? Possibly through the same order.

# Further paths

- Do $\theta$-free, $g \geqslant 5$ graphs have **bounded queue/stack number**? Possibly through the same order.
- Extend the approach to classes forbidding any subgraph of the wall.

# Further paths

- Do $\theta$-free, $g \geqslant 5$ graphs have **bounded queue/stack number**? Possibly through the same order.
- Extend the approach to classes forbidding any subgraph of the wall.

Questions ?

# Further paths

- Do $\theta$-free, $g \geqslant 5$ graphs have **bounded queue/stack number**? Possibly through the same order.
- Extend the approach to classes forbidding any subgraph of the wall.

Questions ?

$\hookrightarrow$ $k$-DS admits no polynomial kernels

# Complexity definitions

**Definition** *Kernel for parameterized problem $\mathcal{Q}$: polytime reduction from $(I, k) \in \mathcal{Q}$ to equivalent $(I', k') \in \mathcal{Q}$.*
*A kernel for parameter $k$ is polynomial if $|I'| = O(k^p)$.*

# Complexity definitions

**Definition** *Kernel for parameterized problem $\mathcal{Q}$: polytime reduction from $(I,k) \in \mathcal{Q}$ to equivalent $(I',k') \in \mathcal{Q}$. A kernel for parameter $k$ is polynomial if $|I'| = O(k^p)$.*

**Definition** VERTEX COVER: *instances $(G,k)$, is there a set $S \subseteq V$ of size at most $k$ covering $E$.*

- ***Connected V-C***: *requires $S$ to be connected in $G$.*

# Complexity definitions

**Definition** *Kernel for parameterized problem $\mathcal{Q}$: polytime reduction from $(I,k) \in \mathcal{Q}$ to equivalent $(I',k') \in \mathcal{Q}$.*
*A kernel for parameter $k$ is polynomial if $|I'| = O(k^p)$.*

**Definition** Vertex Cover*: instances $(G,k)$, is there a set $S \subseteq V$ of size at most $k$ covering $E$.*

- ***Connected V-C****: requires $S$ to be connected in $G$.*

**Definition** $k$-Dominating Set*: Instances $(G,k)$, is there a set $S \subseteq V$ of size at most $k$ dominating $V$.*

# Twin-width : FPT Consequences

**Theorem (Bonnet, Kim, Thomassé, Watrigant '20)**
*FO model checking is FPT with respects to formula size on classes of bounded twin-width given a contraction sequence.*

# Twin-width : FPT Consequences

**Theorem (Bonnet, Kim, Thomassé, Watrigant '20)**
*FO model checking is FPT with respects to formula size on classes of bounded twin-width given a contraction sequence.*

- $k$-IS : $\exists x_1 \ldots \exists x_k \bigwedge_{1 \leqslant i \leqslant j \leqslant k} \neg(x_i = x_j \vee E(x_i, x_j))$,
- $k$-DS : $\exists x_1 \ldots \exists x_k \forall x \bigvee_{1 \leqslant i \leqslant k} (x = x_i) \vee \bigvee_{1 \leqslant i \leqslant k} E(x, x_i)$.

# Twin-width : FPT Consequences

**Theorem (Bonnet, Kim, Thomassé, Watrigant '20)**
*FO model checking is FPT with respects to formula size on classes of bounded twin-width given a contraction sequence.*

- $k$-IS : $\exists x_1 \dots \exists x_k \bigwedge_{1 \leqslant i \leqslant j \leqslant k} \neg(x_i = x_j \vee E(x_i, x_j))$,
- $k$-DS : $\exists x_1 \dots \exists x_k \forall x \bigvee_{1 \leqslant i \leqslant k} (x = x_i) \vee \bigvee_{1 \leqslant i \leqslant k} E(x, x_i)$.

**Theorem (Bonnet, Geniet, Kim, Thomassé, Watrigant '20)**
*$k$-IS admits no polynomial kernel on classes of bounded twin-width unless unless* coNP $\subseteq$ NP/*poly.*

# Twin-width : FPT Consequences

**Theorem (Bonnet, Kim, Thomassé, Watrigant '20)**
*FO model checking is FPT with respects to formula size on classes of bounded twin-width given a contraction sequence.*

- $k$-IS : $\exists x_1 \ldots \exists x_k \bigwedge_{1 \leqslant i \leqslant j \leqslant k} \neg(x_i = x_j \vee E(x_i, x_j))$,
- $k$-DS : $\exists x_1 \ldots \exists x_k \forall x \bigvee_{1 \leqslant i \leqslant k} (x = x_i) \vee \bigvee_{1 \leqslant i \leqslant k} E(x, x_i)$.

**Theorem (Bonnet, Geniet, Kim, Thomassé, Watrigant '20)**
*$k$-IS admits no polynomial kernel on classes of bounded twin-width unless unless* coNP $\subseteq$ NP/*poly*.

$k$-DS admits $O(k^{(t+1)^2})$ kernels notably on sparse ($K_{t,t}$-free) classes (Philip, Raman, Sikdar '12), what about **bounded twin-width**?

# Lower bounds framework

**OR-cross-composition**, from $\mathscr{L}$ to parameterized $\mathscr{Q}$:
Polytime reduction the **OR** of $t$ ("equivalent") instances of $\mathscr{L}$ to one instance of $\mathscr{Q}$.

# Lower bounds framework

**OR-cross-composition**, from $\mathscr{L}$ to parameterized $\mathscr{Q}$:
Polytime reduction the **OR** of $t$ ("equivalent") instances of $\mathscr{L}$ to one instance of $\mathscr{Q}$.

> **Theorem (Bodlaender, Jansen, Kratsch '12)**
> *If an NP-hard language $\mathscr{L}$ admits an OR-cross-composition into a parameterized problem $\mathscr{Q}$, then $\mathscr{Q}$ does not admit a polynomial kernel unless* coNP $\subseteq$ NP*/poly.*

- $\mathscr{L}$ : Tailored $k$-DS (next slide),
- $\mathscr{Q}$ : $k$-DS with twin-width 4.

# Tailored k-DS on grid-like graphs

**Theorem**

$k$-Dominating Set *remains* NP*-hard even when restricted to instances* $(G, k, \mathscr{B} = \{B_1, \ldots, B_k\})$ *such that:*

- $\mathscr{B}$ *partitions* $V(G)$*,* $G$ *has 4-sequence* $\rightarrow G/\mathscr{B}$*,*
- $G/\mathscr{B}$ *is a spanning subgraph of a grid,*

# Tailored k-DS on grid-like graphs

**Theorem**

*$k$-Dominating Set remains* NP*-hard even when restricted to instances* $(G, k, \mathscr{B} = \{B_1, \ldots, B_k\})$ *such that:*

- *$\mathscr{B}$ partitions $V(G)$, $G$ has 4-sequence $\to G/\mathscr{B}$,*
- *$G/\mathscr{B}$ is a spanning subgraph of a grid,*
- *every dominating set of $G$ intersects each $B_i$, for $i \in [k]$*

# Tailored k-DS on grid-like graphs

**Theorem**

*$k$-Dominating Set remains NP-hard even when restricted to instances $(G, k, \mathscr{B} = \{B_1, \ldots, B_k\})$ such that:*

- *$\mathscr{B}$ partitions $V(G)$, $G$ has 4-sequence $\to G/\mathscr{B}$,*
- *$G/\mathscr{B}$ is a spanning subgraph of a grid,*
- *every dominating set of $G$ intersects each $B_i$, for $i \in [k]$*

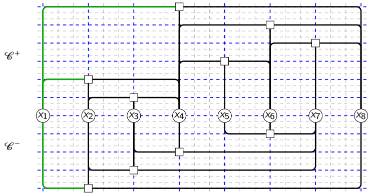



Fig. Reduction sketch

# OR-composition sketch

**Tailored** $k$-**DS**, instances $(I_i)_{i \in [t]} \rightarrow k$-**DS**, $tww \leqslant 4$ instance $(H, k)$

# OR-composition sketch

**Tailored** $k$-**DS**, instances $(I_i)_{i \in [t]} \to k$-**DS**, $tww \leqslant 4$ instance $(H, k)$

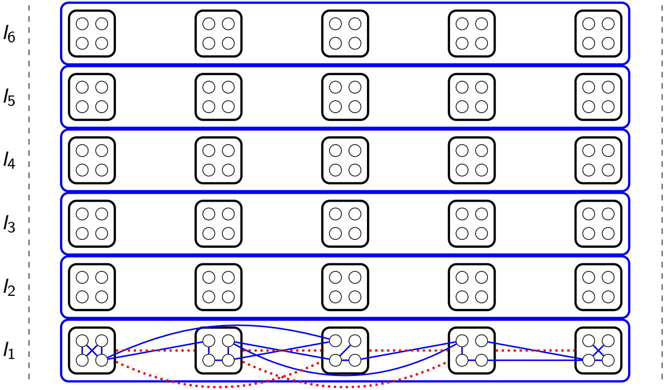



Fig. Instances ↔ layers, partition classes ↔ boxes.

# OR-composition sketch

**Tailored** $k$-**DS**, instances $(I_i)_{i \in [t]}$ → $k$-**DS**, *tww* ⩽ 4 instance $(H, k)$

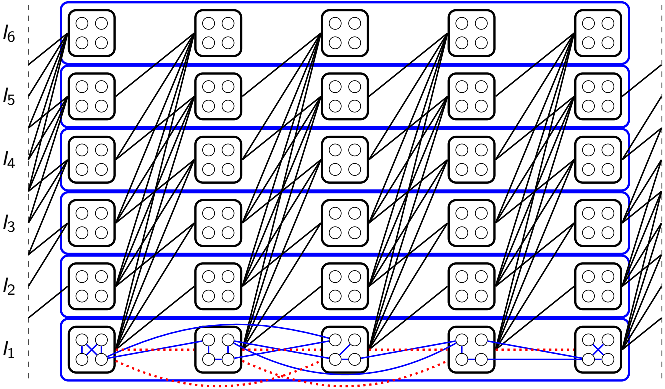



Fig. Instances ↔ layers, partition classes ↔ boxes.
**In** $H$**:** Top instance ↔ dummy, half graphs cycle between classes

# Positive Tailored instance ⇒ Positive composition

Assume one instance ($I_4$ here) is positive, pick its solution in $H$:

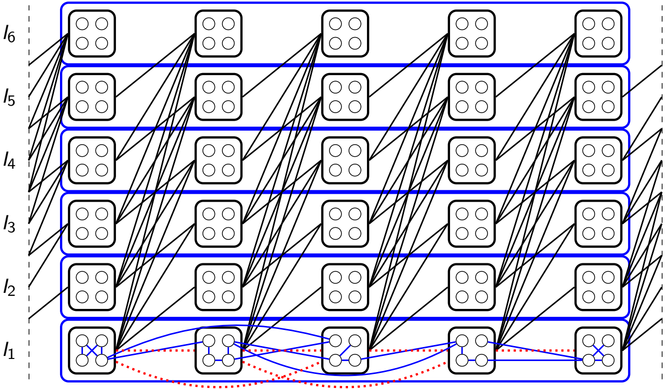



Fig. $t$ Instances ↔ layers, $k$ partition classes ↔ boxes

# Positive Tailored instance ⇒ Positive composition

Assume one instance ($I_4$ here) is positive, pick its solution in $H$:

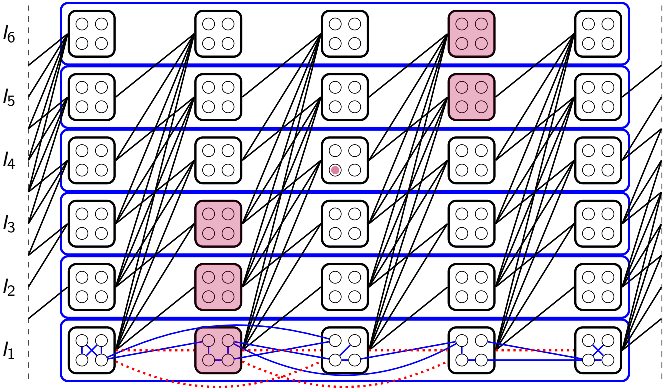



Fig. $t$ Instances ↔ layers, $k$ partition classes ↔ boxes

# Positive Tailored instance ⇒ Positive composition

Assume one instance ($I_4$ here) is positive, pick its solution in $H$:

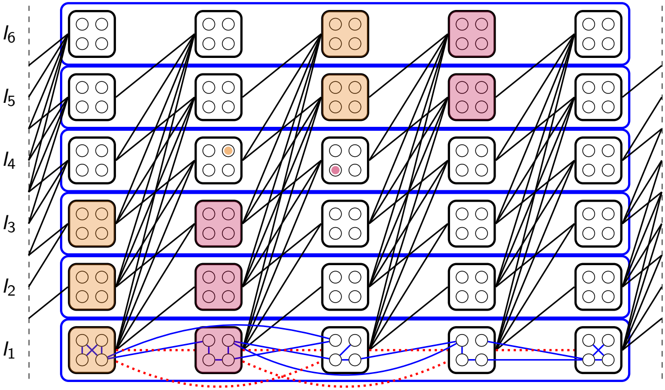



Fig. $t$ Instances ↔ layers, $k$ partition classes ↔ boxes

# Positive Tailored instance ⇒ Positive composition

Assume one instance ($I_4$ here) is positive, pick its solution in $H$:

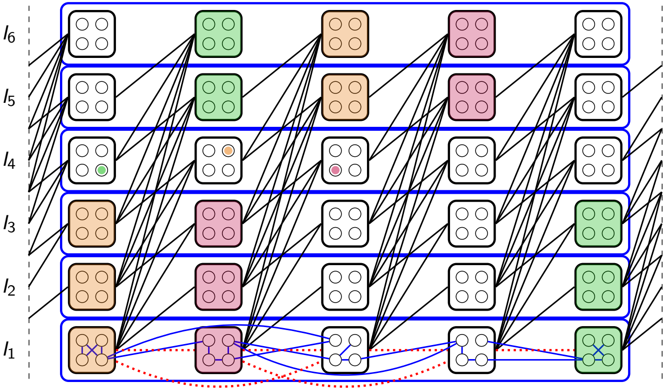



Fig. $t$ Instances ↔ layers, $k$ partition classes ↔ boxes

# Positive Tailored instance ⇒ Positive composition

Assume one instance ($I_4$ here) is positive, pick its solution in $H$:

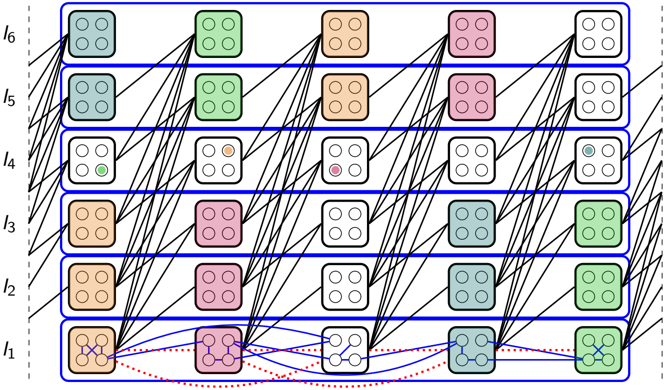



Fig. $t$ Instances ↔ layers, $k$ partition classes ↔ boxes

# Positive Tailored instance ⇒ Positive composition

Assume one instance ($I_4$ here) is positive, pick its solution in $H$:

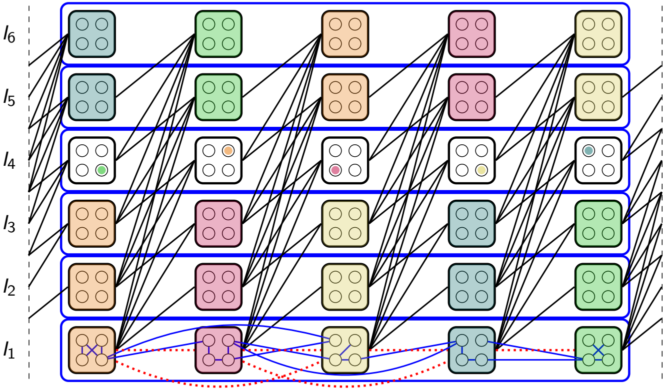
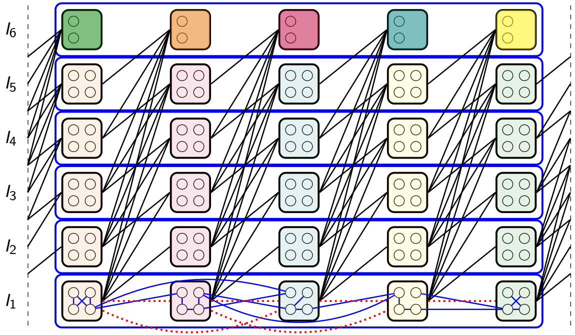



Fig. $t$ Instances ↔ layers, $k$ partition classes ↔ boxes

# Positive composition ⇒ Positive tailored instance

Assume $(H, k)$ is positive:

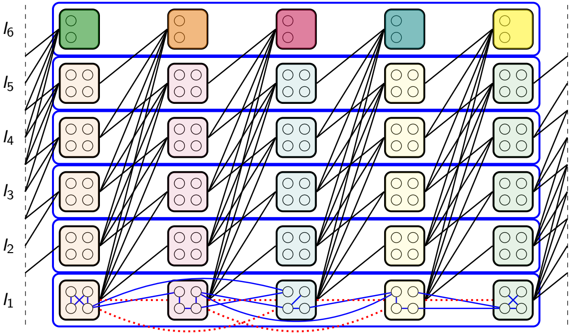
- Dummy instance $I_6$ forces one vertex per column (figure),



# Positive composition ⇒ Positive tailored instance

Assume $(H, k)$ is positive:

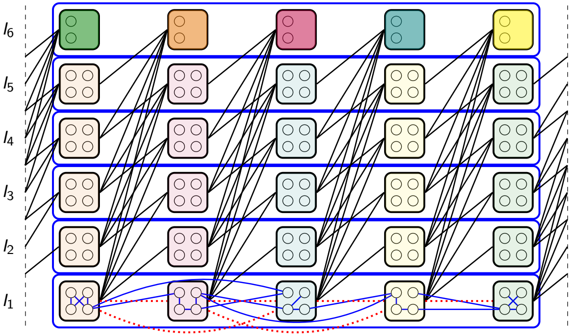
- Dummy instance $I_6$ forces one vertex per column (figure),
- Ideally : all choices on a single layer $i$ → positive $I_i$,



# Positive composition ⇒ Positive tailored instance

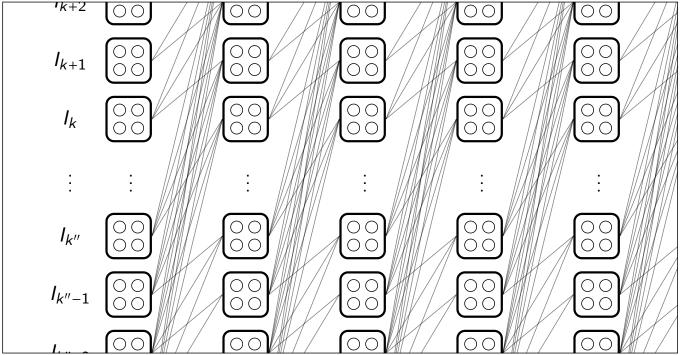
Assume $(H, k)$ is positive:

- Dummy instance $I_6$ forces one vertex per column (figure),
- Ideally : all choices on a single layer $i$ → positive $I_i$,
- Assume not...



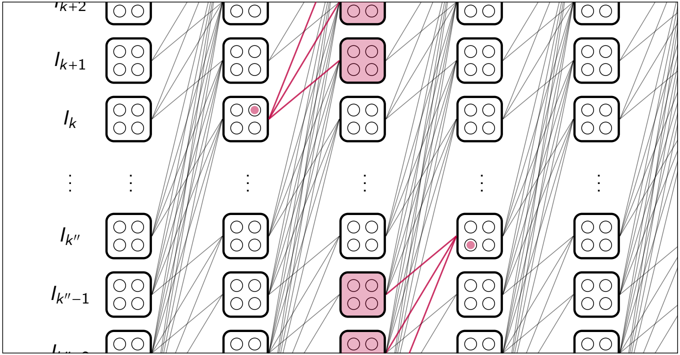
# Positive composition ⇒ Positive tailored instance

- Non-identical layer choices → domination gap,



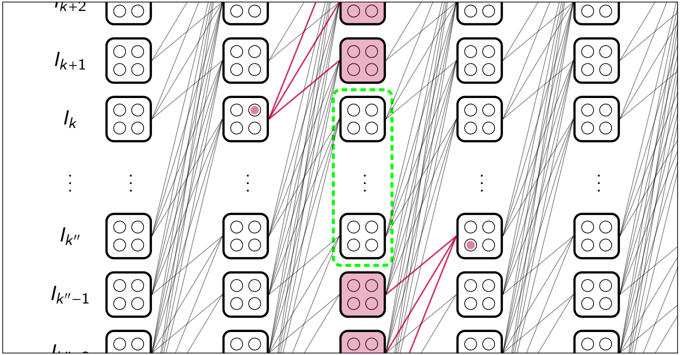
# Positive composition ⇒ Positive tailored instance

- Non-identical layer choices → domination gap,
- At least two classes are not dominated by neighbouring columns,



# Positive composition ⇒ Positive tailored instance

- Non-identical layer choices → domination gap,
- At least two classes are not dominated by neighbouring columns,
- Only one choice left in this column → absurd.



# Bounding the twin-width

- For any $I_i$, each $\mathscr{B}_{i,j}$ is a module in $H - I_i$ → contraction impacts only $I_i$,

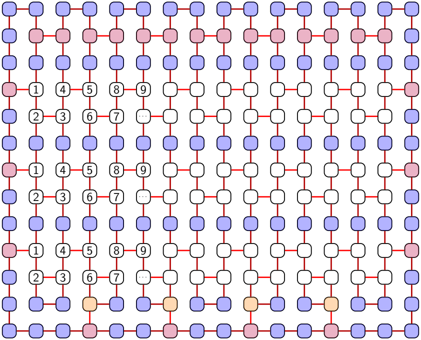



Fig. An instance after contracting classes.

# Bounding the twin-width

- For any $I_i$, each $\mathscr{B}_{i,j}$ is a module in $H - I_i$ → contraction impacts only $I_i$,
- Contract each partition class of each instance

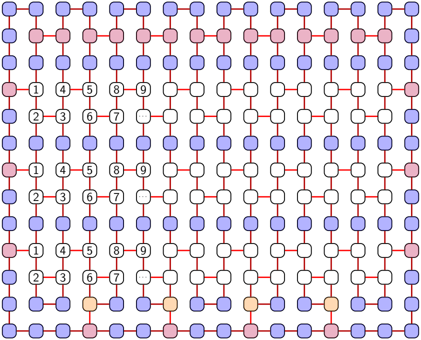



Fig. An instance after contracting classes.

# Bounding the twin-width

- For any $I_i$, each $\mathscr{B}_{i,j}$ is a module in $H - I_i$ → contraction impacts only $I_i$,
- Contract each partition class of each instance
- Result: red subgraph of grid (altered for *tww* 4).

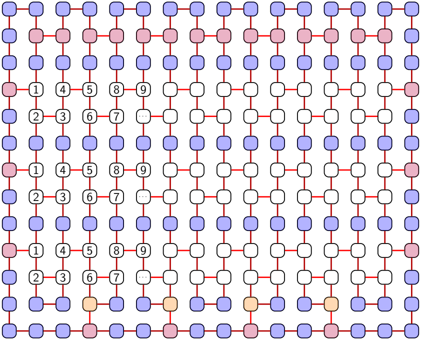
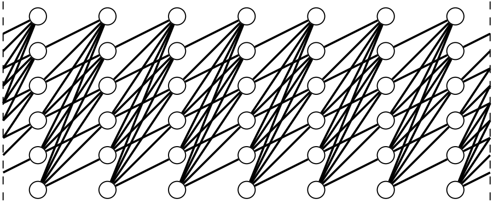



Fig. An instance after contracting classes.

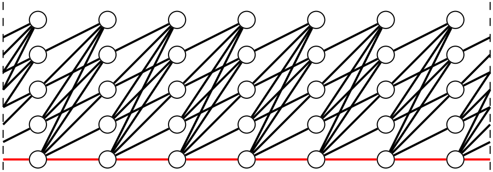
# Bounding the twin-width

- Between instances : cycle of half graphs,



# Bounding the twin-width

- Between instances : cycle of half graphs,
- Half graphs cycles have tww 3 : successively contract bottommost two layers



# Bounding the twin-width

- In the Composition: half-graph cycle between "homogeneous" vertices, (at the same position in their respective instance).

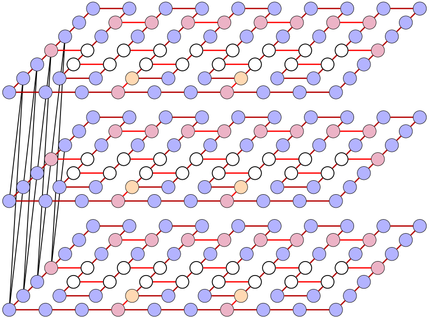


Fig. Instances and the half-graph cycle

# Bounding the twin-width

- In the Composition: half-graph cycle between "homogeneous" vertices, (at the same position in their respective instance).
- Goal: Successively contract the bottommost two instances.

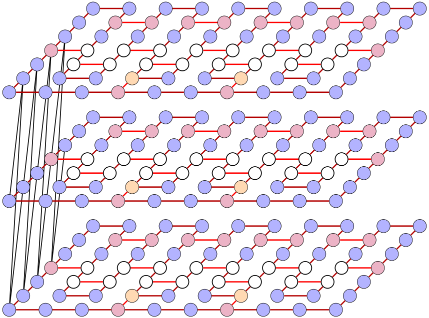


Fig. Instances and the half-graph cycle

# Bounding the twin-width

- In the Composition: half-graph cycle between "homogeneous" vertices, (at the same position in their respective instance).
- Goal: Successively contract the bottommost two instances.
- Each contraction: creates red edges only locally → induction

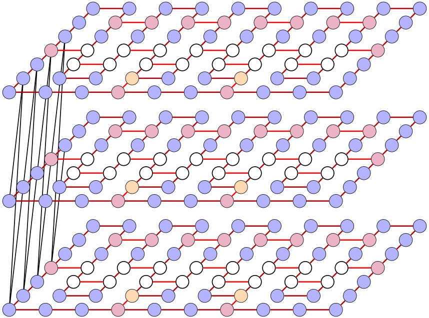


Fig. Instances and the half-graph cycle

# Concluding

- NP-hard Tailored $k$-DS OR-cross-composes into $k$-DS,

# Concluding

- NP-hard Tailored $k$-DS OR-cross-composes into $k$-DS,
- The composed instance has tww at most four.

# Concluding

- NP-hard Tailored $k$-DS OR-cross-composes into $k$-DS,
- The composed instance has tww at most four.

**Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*Unless* coNP $\subseteq$ NP/*poly, $k$-*DOMINATING SET *on graphs of twin-width at most 4 does not admit a polynomial kernel, even if a 4-sequence of the graph is given.*

# A (sub)quadratic kernel for C-VC on VC-density 1

**Lemma (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*There exists $f$ s.t. for any $G$ of tww( $d$) and $X \subseteq V(G)$: number of distinct neighborhoods in $X$ is at most $f(d)|X|$.*

# A (sub)quadratic kernel for C-VC on VC-density 1

**Lemma (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*There exists $f$ s.t. for any $G$ of tww( $d$) and $X \subseteq V(G)$: number of distinct neighborhoods in $X$ is at most $f(d)|X|$.*

**Kernel pre-processing:**

- Bounded tww $G$: take $X$ 2-approx for VC,

# A (sub)quadratic kernel for C-VC on VC-density 1

**Lemma (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*There exists $f$ s.t. for any $G$ of tww( $d$) and $X \subseteq V(G)$: number of distinct neighborhoods in $X$ is at most $f(d)|X|$.*

**Kernel pre-processing:**

- Bounded tww $G$: take $X$ 2-approx for VC,
- if $|X| \geqslant 2k+1$ : no solution,

# A (sub)quadratic kernel for C-VC on VC-density 1

**Lemma (Bonnet, Kim, R., Thomassé, Watrigant '21)**
*There exists $f$ s.t. for any $G$ of tww( $d$) and $X \subseteq V(G)$: number of distinct neighborhoods in $X$ is at most $f(d)|X|$.*

**Kernel pre-processing:**

- Bounded tww $G$: take $X$ 2-approx for VC,
- if $|X| \geqslant 2k+1$ : no solution,
- $\rightarrow$ in $X$, number of distinct neighbourhoods $\leqslant f(d)k$

# A (sub)quadratic kernel for C-VC on VC-density 1

**Reduction rule:** If there is $S \subseteq V(G) \setminus X$ with identical neighbourhood in $X$ and $|S| > k$, delete a vertex of $S$.

# A (sub)quadratic kernel for C-VC on VC-density 1

**Reduction rule:** If there is $S \subseteq V(G) \setminus X$ with identical neighbourhood in $X$ and $|S| > k$, delete a vertex of $S$.

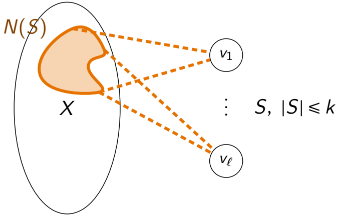



Fig. Resulting instance: $(G', k)$

# A (sub)quadratic kernel for C-VC on VC-density 1

**Reduction rule:** If there is $S \subseteq V(G) \setminus X$ with identical neighbourhood in $X$ and $|S| > k$, delete a vertex of $S$.

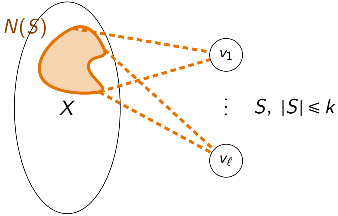



Fig. Resulting instance: $(G', k)$

If $(G, k)$ has a solution, replace deletion with twin → reconnects $T$

# A (sub)quadratic kernel for C-VC on VC-density 1

**Reduction rule:** If there is $S \subseteq V(G) \setminus X$ with identical neighbourhood in $X$ and $|S| > k$, delete a vertex of $S$.

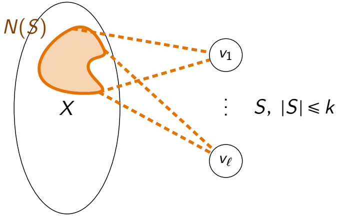



Fig. Resulting instance: $(G', k)$

If $(G, k)$ has a solution, replace deletion with twin → reconnects $T$

If $(G', k)$ has a solution $T$, $T$ is also a solution for $(G, k)$,

# A (sub)quadratic kernel for C-VC on VC-density 1

**Reduction rule:** If there is $S \subseteq V(G) \setminus X$ with identical neighbourhood in $X$ and $|S| > k$, delete a vertex of $S$.

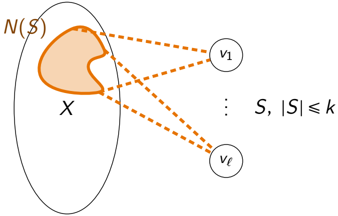



Fig. Resulting instance: $(G', k)$

If $(G,k)$ has a solution, replace deletion with twin → reconnects $T$

If $(G',k)$ has a solution $T$, $T$ is also a solution for $(G,k)$,

- Take deleted $s$, $S \backslash s \not\subseteq T$ as ind. set with $|S| \geqslant k$,
- Therefore $N(S) \subseteq T$, and $s$ is covered by $T$ in $G$.

## Concluding

- Applying the reduction yields an equivalent instance,

# Concluding

- Applying the reduction yields an equivalent instance,
- The kernel has size at most $f(d) * k^2$,

# Concluding

- Applying the reduction yields an equivalent instance,
- The kernel has size at most $f(d) * k^2$,

# Concluding

- Applying the reduction yields an equivalent instance,
- The kernel has size at most $f(d) * k^2$,

> **Theorem (Bonnet, Kim, R., Thomassé, Watrigant '21)**
> *On classes of VC-density 1,* Connected *$k$*-Vertex Cover *admits kernels with $O(k^2)$ vertices (and even $O(k^{1.5})$).*

# Further Directions

- Does $k$-DS on tww $\leqslant 3$ admit polynomial kernels?

# Further Directions

- Does $k$-DS on tww $\leqslant 3$ admit polynomial kernels?
- Is there a linear kernel for Connected-VC?

# Further Directions

- Does $k$-DS on tww $\leqslant 3$ admit polynomial kernels?
- Is there a linear kernel for Connected-VC?

Thank you!

# Bibliography

V. Dujmovi, D. Eppstein, R. Hickingbotham, P. Morin, and D. R. Wood. "Stack-number is not bounded by queue-number". In: (2021). arXiv: 2011.04195 [math.CO]

N. L. D. Sintiari and N. Trotignon. "(Theta, triangle)-free and (even hole, $K_4$)-free graphs. Part 1 : Layered wheels". In: *CoRR* abs/1906.10998 (2019). arXiv: 1906.10998. URL: http://arxiv.org/abs/1906.10998

T. Abrishami, M. Chudnovsky, S. Hajebi, and S. Spirkl. "Induced subgraphs and tree decompositions III. Three-path-configurations and logarithmic treewidth". In: (2021). arXiv: 2109.01310 [math.CO]

Édouard Bonnet, E. J. Kim, S. Thomassé, and R. Watrigant. "Twin-width I: tractable FO model checking". In: (2020). arXiv: 2004.14789 [cs.DS]

# Bibliography

Édouard Bonnet, C. Geniet, E. J. Kim, S. Thomassé, and R. Watrigant. "Twin-width II: small classes". In: (2020). arXiv: 2006.09877 [cs.DM]

H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. "Kernelization Lower Bounds By Cross-Composition". In: (2012). arXiv: 1206.5941 [cs.CC]

G. Philip, V. Raman, and S. Sikdar. "Polynomial Kernels for Dominating Set in Graphs of Bounded Degeneracy and Beyond". In: *ACM Trans. Algorithms* 9.1 (Dec. 2012). ISSN: 1549-6325. DOI: 10.1145/2390176.2390187. URL: https://doi.org/10.1145/2390176.2390187

Édouard Bonnet, E. J. Kim, A. Reinald, S. Thomassé, and R. Watrigant. "Twin-width and polynomial kernels". In: (2021). arXiv: 2107.02882 [cs.DS]