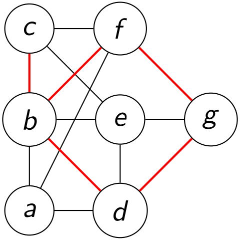
Twin-width delineation and win-wins

Édouard Bonnet

ENS Lyon, LIP

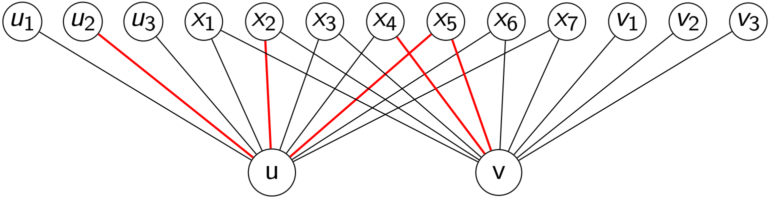
July 4th, 2022, Paris

Trigraphs



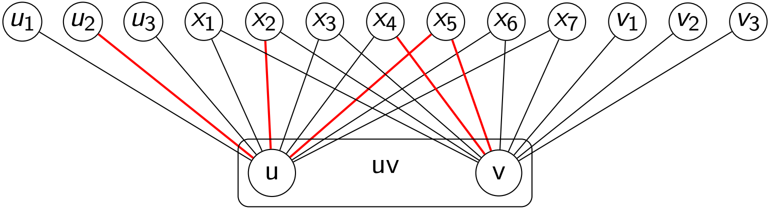
Three outcomes between a pair of vertices:
edge, or non-edge, or red edge (error edge)

Contractions in trigraphs



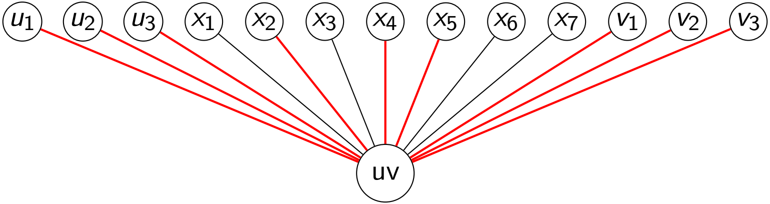
Identification of two non-necessarily adjacent vertices

Contractions in trigraphs



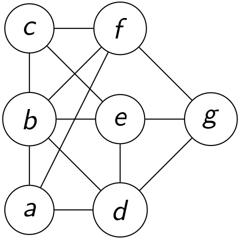
Identification of two non-necessarily adjacent vertices

Contractions in trigraphs



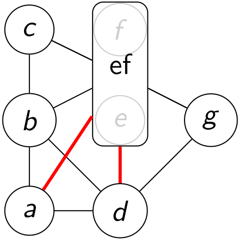
edges to $N(u) \triangle N(v)$ turn red, for $N(u) \cap N(v)$ red is absorbing

Contraction sequence



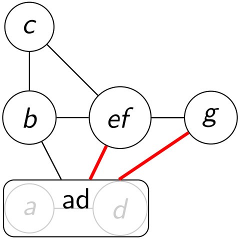
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \dots, G_2, G_1$ such that
G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



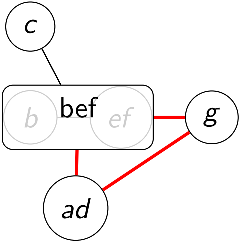
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



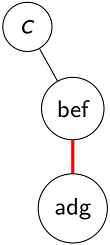
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



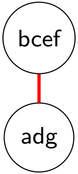
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \dots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



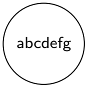
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \dots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

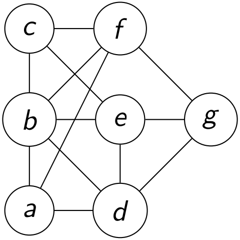
Contraction sequence



A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Twin-width

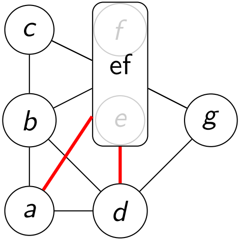
tww(G): Least integer d such that G admits a contraction sequence where all trigraphs have *maximum red degree* at most d.



Maximum red degree $= 0$
overall maximum red degree = 0

Twin-width

tww(G): Least integer d such that G admits a contraction sequence where all trigraphs have *maximum red degree* at most d.

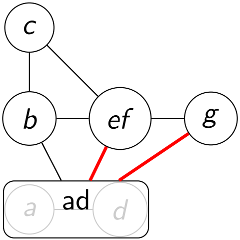


Maximum red degree $= 2$

overall maximum red degree = 2

Twin-width

tww(G): Least integer d such that G admits a contraction sequence where all trigraphs have *maximum red degree* at most d.

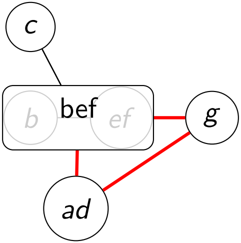


Maximum red degree $= 2$

overall maximum red degree = 2

Twin-width

tww(G): Least integer d such that G admits a contraction sequence where all trigraphs have *maximum red degree* at most d.

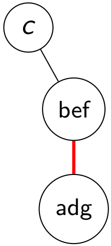


Maximum red degree = 2

overall maximum red degree = 2

Twin-width

tww(G): Least integer d such that G admits a contraction sequence where all trigraphs have *maximum red degree* at most d.

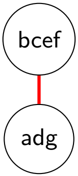


Maximum red degree $= 1$

overall maximum red degree = 2

Twin-width

tww(G): Least integer d such that G admits a contraction sequence where all trigraphs have *maximum red degree* at most d.

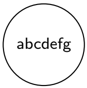


Maximum red degree $= 1$

overall maximum red degree = 2

Twin-width

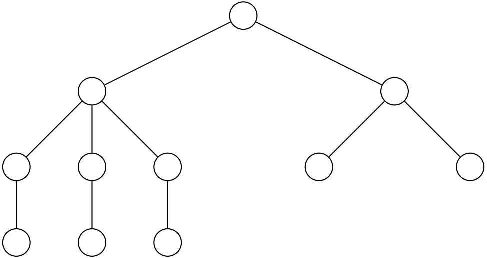
tww(G): Least integer d such that G admits a contraction sequence where all trigraphs have *maximum red degree* at most d.



Maximum red degree $= 0$

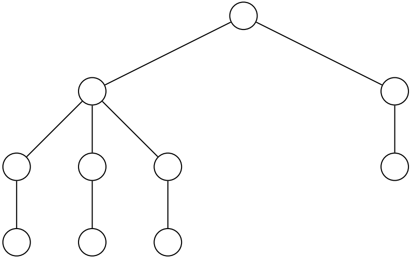
overall maximum red degree = 2

Trees have twin-width at most 2



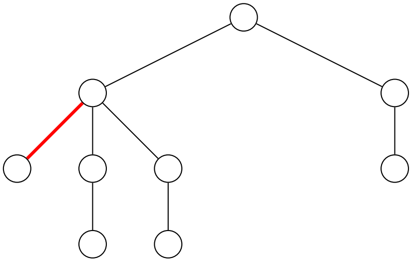
If possible, contract two twin leaves

Trees have twin-width at most 2



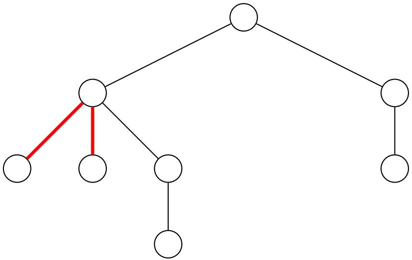
If not, contract a deepest leaf with its parent

Trees have twin-width at most 2



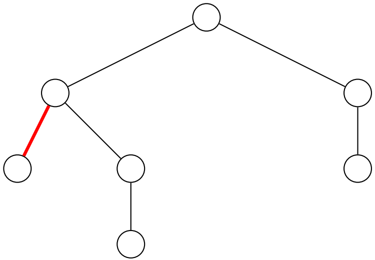
If not, contract a deepest leaf with its parent

Trees have twin-width at most 2



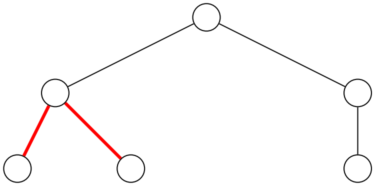
If possible, contract two twin leaves

Trees have twin-width at most 2



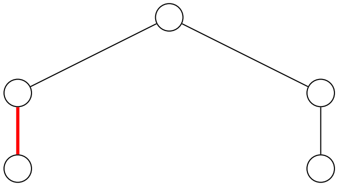
Cannot create a red degree-3 vertex

Trees have twin-width at most 2



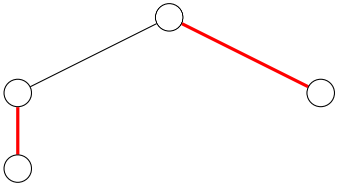
Cannot create a red degree-3 vertex

Trees have twin-width at most 2



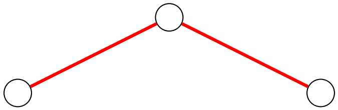
Cannot create a red degree-3 vertex

Trees have twin-width at most 2



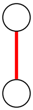
Cannot create a red degree-3 vertex

Trees have twin-width at most 2



Cannot create a red degree-3 vertex

Trees have twin-width at most 2

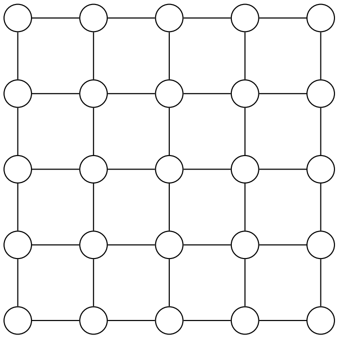


Cannot create a red degree-3 vertex

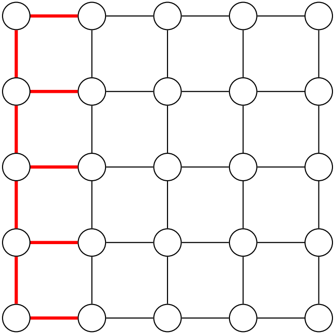
Trees have twin-width at most 2

Generalization to bounded *treewidth* and even bounded *rank-width*

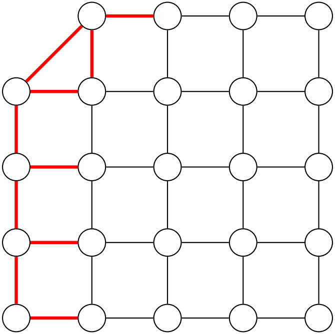
Grids have twin-width at most 4



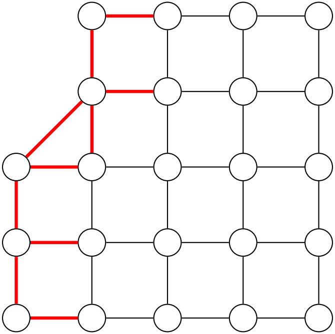
Grids have twin-width at most 4



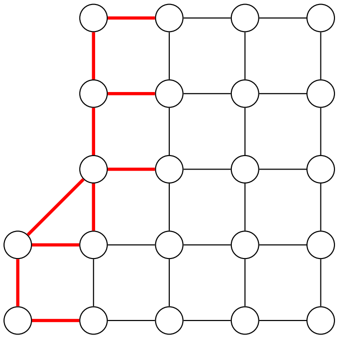
Grids have twin-width at most 4



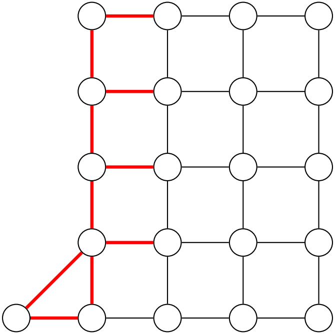
Grids have twin-width at most 4



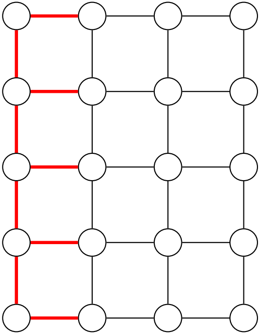
Grids have twin-width at most 4



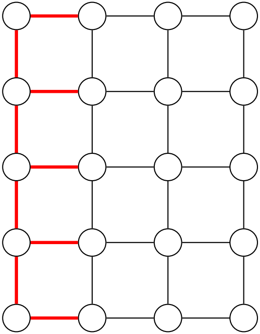
Grids have twin-width at most 4



Grids have twin-width at most 4

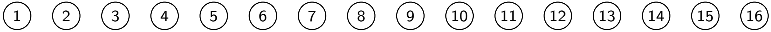


Grids have twin-width at most 4

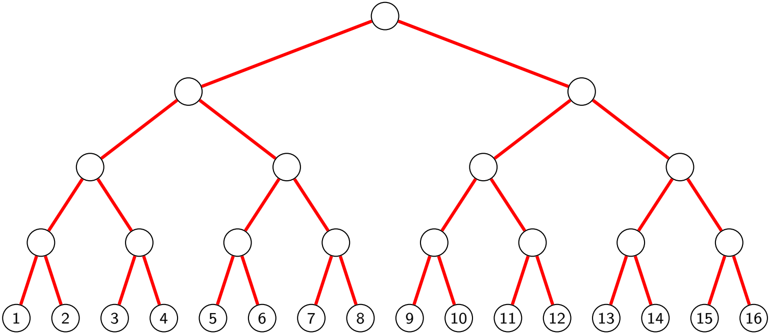


More generally, d-dimensional grids have twin-width $\Theta(d)$

($\geqslant 2 \log n$)-subdivisions have twin-width at most 4

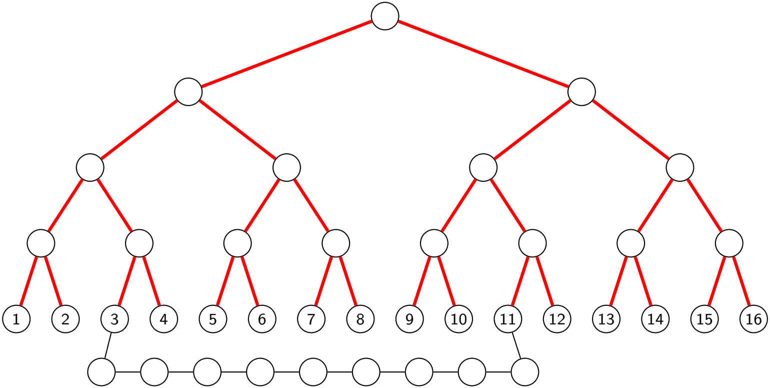


($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



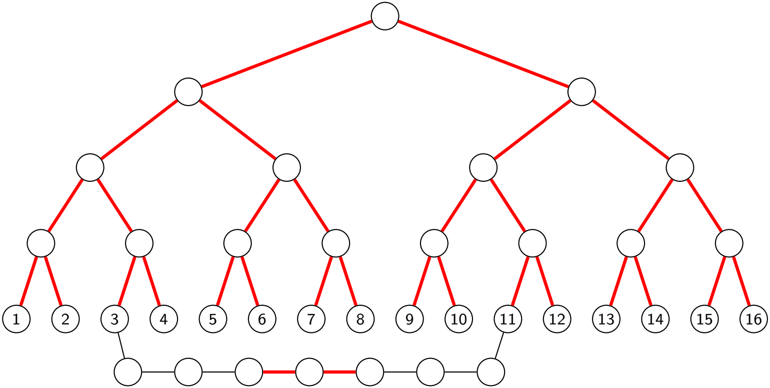
Add a red full binary tree whose leaves are the vertex set

($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



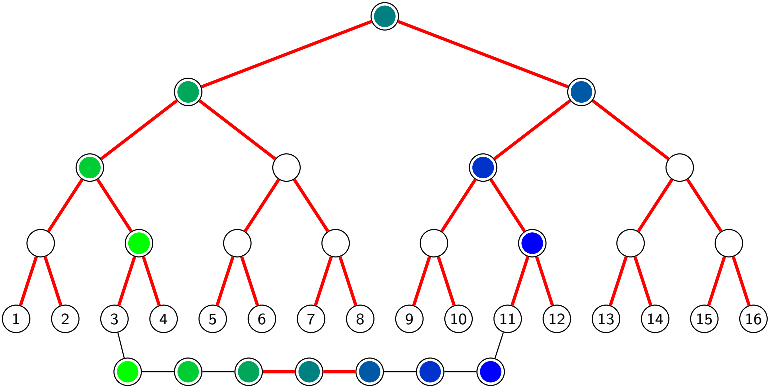
Take any subdivided edge

($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



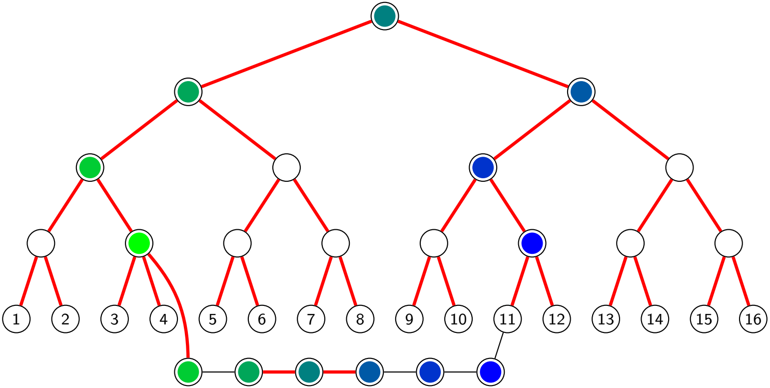
Shorten it to the length of the path in the red tree

($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



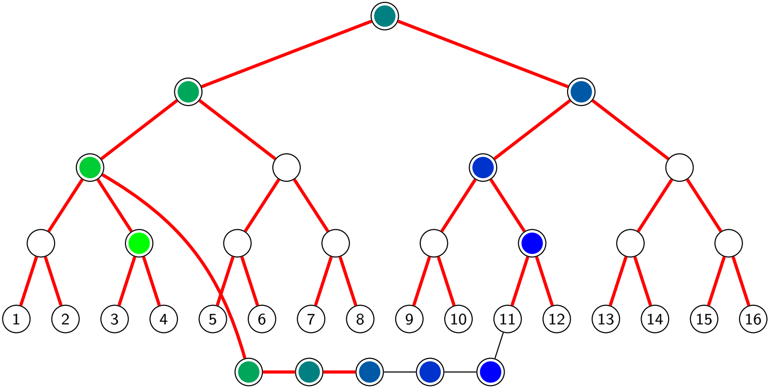
Zip the subdivided edge in the tree

($\geqslant 2\log n$)-subdivisions have twin-width at most 4



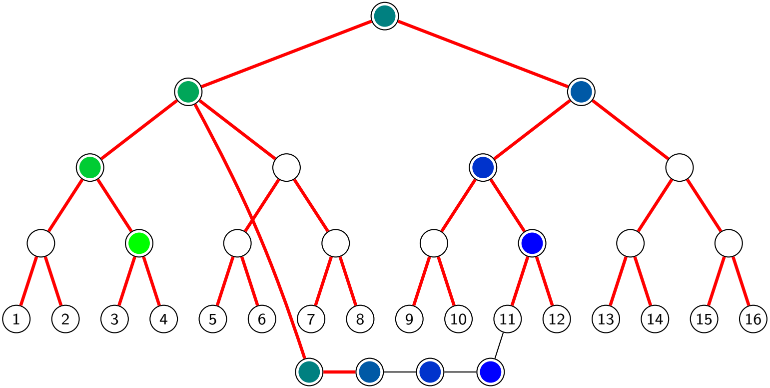
Zip the subdivided edge in the tree

($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



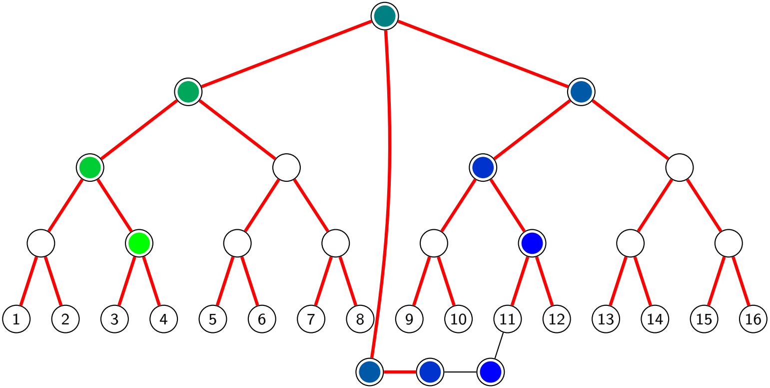
Zip the subdivided edge in the tree

($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



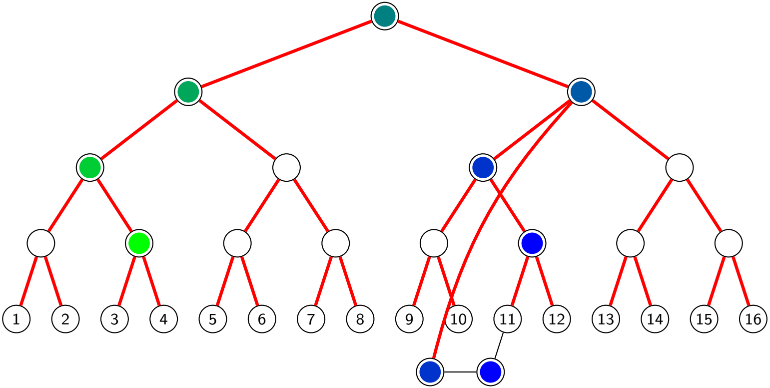
Zip the subdivided edge in the tree

($\geqslant 2\log n$)-subdivisions have twin-width at most 4



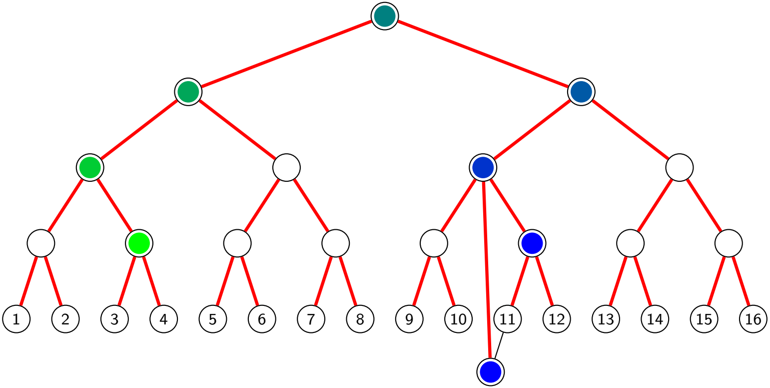
Zip the subdivided edge in the tree

$(\geqslant 2\log n)$-subdivisions have twin-width at most 4



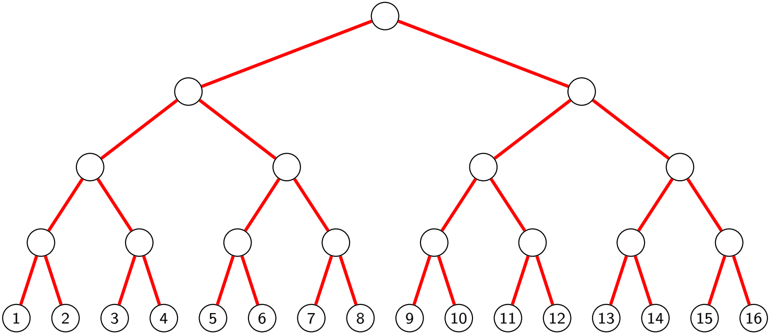
Zip the subdivided edge in the tree

($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



Zip the subdivided edge in the tree

$(\geqslant 2 \log n)$-subdivisions have twin-width at most 4



Move to the next subdivided edge

Theorem (B., Geniet, Kim, Thomassé, Watrigant '20, '21)

The following classes have bounded twin-width, and $O(1)$-sequences can be computed in polynomial time.

- *Bounded rank-width or clique-width graphs,*
- *every hereditary proper subclass of permutation graphs,*
- *posets of bounded antichain size,*
- *unit interval graphs,*
- *K_t-minor free graphs,*
- *map graphs,*
- *subgraphs of d-dimensional grids,*
- *K_t-free unit d-dimensional ball graphs,*
- *$\Omega(\log n)$-subdivisions of all the n-vertex graphs,*
- *cubic expanders defined by iterative random 2-lifts from K_4,*
- *strong products of two bounded twin-width classes, one with bounded degree, etc.*

Theorem (B., Geniet, Kim, Thomassé, Watrigant '20, '21)

The following classes have bounded twin-width, and $O(1)$-sequences can be computed in polynomial time.

- *Bounded rank-width or clique-width graphs,*
- *every hereditary proper subclass of permutation graphs,*
- *posets of bounded antichain size,*
- *unit interval graphs,*
- *K_t-minor free graphs,*
- *map graphs,*
- *subgraphs of d-dimensional grids,*
- *K_t-free unit d-dimensional ball graphs,*
- *$\Omega(\log n)$-subdivisions of all the n-vertex graphs,*
- *cubic expanders defined by iterative random 2-lifts from K_4,*
- *strong products of two bounded twin-width classes, one with bounded degree, etc.*

Do contraction sequences allow for faster algorithms?

First-order model checking on graphs

GRAPH FO MODEL CHECKING	**Parameter:** $\lvert\varphi\rvert$

Input: A graph G and a first-order sentence $\varphi \in FO(\{E\})$

Question: $G \models \varphi$?

First-order model checking on graphs

Graph FO Model Checking	**Parameter:** $\|\varphi\|$
Input: A graph G and a first-order sentence $\varphi \in FO(\{E\})$	
Question: $G \models \varphi$?	

Example:

$$\varphi = \exists x_1 \exists x_2 \cdots \exists x_k \forall x \forall y \ (E(x, y) \Rightarrow \bigvee_{1 \leqslant i \leqslant k} x = x_i \vee y = x_i)$$

$G \models \varphi$? $\Leftrightarrow$ k-Vertex Cover

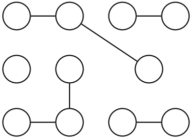
FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete



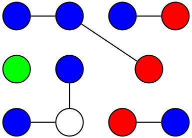
FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula

$\varphi(x, y) = \neg E(x, y)$ (complement)

$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete



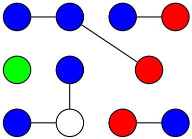
FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula

$\varphi(x, y) = \neg E(x, y)$ (complement)

$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete



$$\varphi(x, y) = E(x, y) \vee (G(x) \wedge B(y) \wedge \neg\exists z R(z) \wedge E(y, z))$$
$$\vee(R(x) \wedge B(y) \wedge \exists z R(z) \wedge E(y, z) \wedge \neg\exists z B(z) \wedge E(y, z))$$

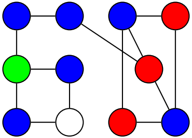
FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula

$\varphi(x, y) = \neg E(x, y)$ (complement)

$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete

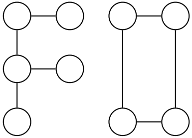


$$\varphi(x, y) = E(x, y) \vee (G(x) \wedge B(y) \wedge \neg\exists z R(z) \wedge E(y, z))$$
$$\vee (R(x) \wedge B(y) \wedge \exists z R(z) \wedge E(y, z) \wedge \neg\exists z B(z) \wedge E(y, z))$$

FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

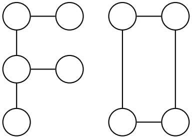
FO transduction: color by $O(1)$ unary relations, interpret, delete



FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete



Theorem (B., Kim, Thomassé, Watrigant '20)

Any FO transduction of a bounded twin-width class has bounded twin-width.

Dependence and monadic dependence

A class $\mathscr{C}$ is
dependent, if the hereditary closure of every simple interpretation of $\mathscr{C}$ misses some graph
monadically dependent, if every transduction of $\mathscr{C}$ misses some graph [Baldwin, Shelah '85]

Dependence and monadic dependence

A class $\mathscr{C}$ is
dependent, if the hereditary closure of every simple interpretation of $\mathscr{C}$ misses some graph
monadically dependent, if every transduction of $\mathscr{C}$ misses some graph [Baldwin, Shelah '85]

Theorem (Downey, Fellows, Taylor '96)

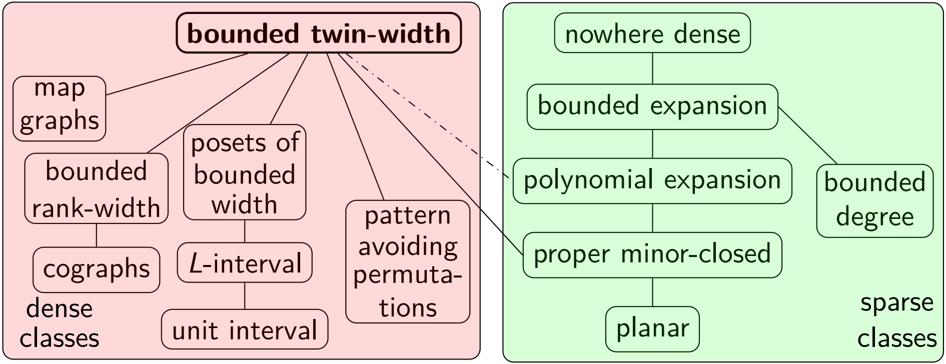
FO model checking is $AW[*]$-complete on general graphs,
thus unlikely FPT on independent classes.

Tractable: FO model checking is FPT on the class

Conjecture (FO, Workshop in Warwick '16, Gajarský et al. '18)

Every monadically dependent class is tractable, with equivalence among hereditary classes.

Tractable classes



Theorem (B., Kim, Thomassé, Watrigant '20)

FO MODEL CHECKING *solvable in* $f(|\varphi|, d)n$ *on graphs with a d-sequence.*

Delineation

$\mathcal{D}$ is **delineated** if for every hereditary $\mathcal{C} \subseteq \mathcal{D}$,
$\mathcal{C}$ has bounded twin-width $\Leftrightarrow \mathcal{C}$ is monadically dependent

Delineation

$\mathcal{D}$ is **delineated** if for every hereditary closure $\mathcal{C}$ of a subclass of $\mathcal{D}$, $\mathcal{C}$ has bounded twin-width $\Leftrightarrow \mathcal{C}$ is monadically dependent.

Delineation

$\mathcal{D}$ is **delineated** if for every hereditary closure $\mathcal{C}$ of a subclass of $\mathcal{D}$, $\mathcal{C}$ has bounded twin-width $\Leftrightarrow \mathcal{C}$ is monadically dependent.

$\mathcal{D}$ is **effectively delineated** if further twin-width is FPT approximable in $\mathcal{D}$

Delineation

$\mathcal{D}$ is **delineated** if for every hereditary closure $\mathcal{C}$ of a subclass of $\mathcal{D}$, $\mathcal{C}$ has bounded twin-width $\Leftrightarrow \mathcal{C}$ is monadically dependent.

$\mathcal{D}$ is **effectively delineated** if further twin-width is FPT approximable in $\mathcal{D}$

Observation

Assuming $FPT \neq AW[*]$, for every hereditary subclass $\mathcal{C}$ of an effectively delineated class:
FO model checking is FPT on $\mathcal{C} \Leftrightarrow \mathcal{C}$ has bounded twin-width.

The FO conjecture is settled on subclasses of delineated classes

How hard is computing twin-width?

Theorem (Bergé, B., Déprés '22)

It is NP-complete to decide if the twin-width is at most 4.

How hard is computing twin-width?

Theorem (Bergé, B., Déprés '22)

It is NP-complete to decide if the twin-width is at most 4.

Question

Is there an FPT $f(OPT)$-approximation of twin-width?

Question

Is twin-width at most k polytime recognizable? (for $k \in \{2, 3\}$)

Grid number, mixed number

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

4-grid minor

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

3-mixed minor

$\mathrm{gn}(M) =$ largest k such that M has a k-grid minor
$\mathrm{mxn}(M) =$ largest k such that M has a k-mixed minor

Grid number, mixed number

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

4-grid minor

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

3-mixed minor

$\text{gn}(G) = \min\ \text{gn}(M)$ among every adjacency matrix M of G

$\text{mxn}(G) = \min\ \text{mxn}(M)$ among every adjacency matrix M of G

Twin-width and mixed/grid number

Theorem (B., Kim, Watrigant, Thomassé '20)

For every graph G, $\frac{mxn(G)-1}{2} \leqslant tww(G) \leqslant 2^{2^{O(mxn(G))}}$.

Corollary

For every graph G, $tww(G) \leqslant 2^{O(gn(G))}$.

Twin-width and mixed/grid number

Theorem (B., Kim, Watrigant, Thomassé '20)

For every graph G, $\frac{mxn(G)-1}{2} \leqslant tww(G) \leqslant 2^{2^{O(mxn(G))}}$.

Corollary

For every graph G, $tww(G) \leqslant 2^{O(gn(G))}$.

Theorem (B., Déprés '22)

$\forall c < 1$, $\exists$ *a class* $\mathcal{C}$ *of unbounded twin-width such that* $\forall G \in \mathcal{C}$,

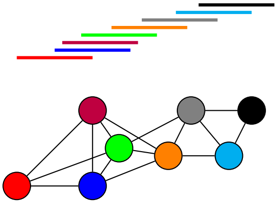
$$tww(G) > 2^{c \cdot (gn(G)-2)}.$$

Question

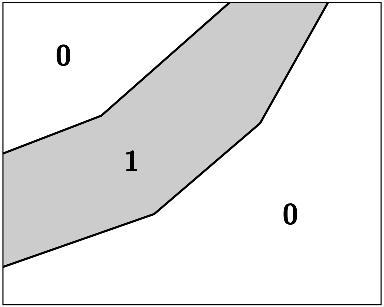
Is the double-exponential dependence in mixed number necessary?

Unit interval graphs

Intersection graph of unit segments on the real line

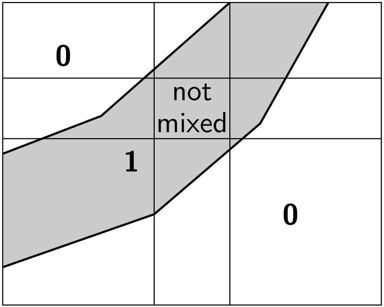


Unit interval graphs have bounded twin-width



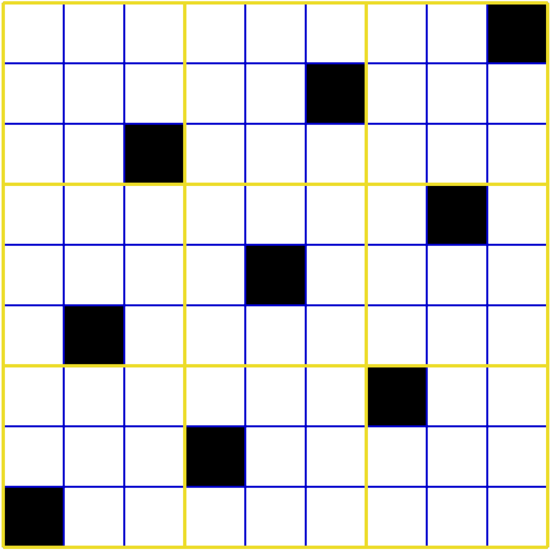
order by left endpoints

Unit interval graphs have bounded twin-width



No 3-by-3 grid has all 9 cells crossed by two non-decreasing curves

Regularizing mixed minors, k-grid permutation

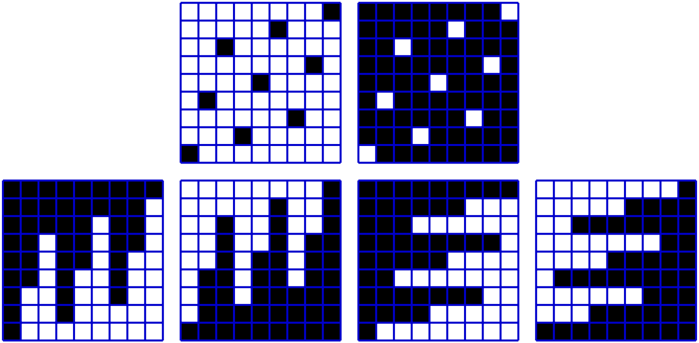


Here with $k = 3$, it has every 3-permutation as subpermutation

The 6 universal patterns of unbounded twin-width

Theorem (B., Giocanti, Ossona de Mendez, Simon, Thomassé, Toruńczyk '22)

$\exists f$ s.t. all the adjacency matrices of a graph of twin-width $\geqslant f(k)$ contains a k-grid permutation submatrix or one of its 5 encodings



Semi-induced matching, antimatching, and 4 half-graphs or ladders

Effectively delineated classes

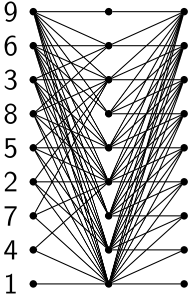
Ordered graphs, permutation graphs, **interval graphs**, etc.

Effectively delineated classes

Ordered graphs, permutation graphs, **interval graphs**, etc.

Find a natural ordering of the vertex set

- no universal pattern $\to$ bounded twin-width
- universal pattern $\to$ "transversal pair," witness of monadic independence



transversal pair: encoding of the *ordered* grid permutation

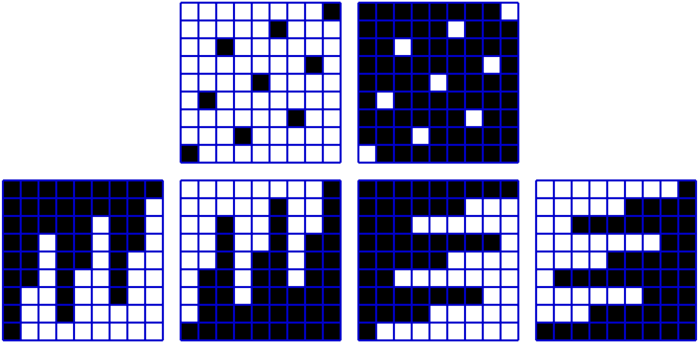
Ordered graphs are delineated

Simply order along the linear order of the binary structure

Ordered graphs are delineated

Simply order along the linear order of the binary structure

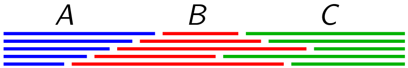
Either witnesses bounded twin-width, or



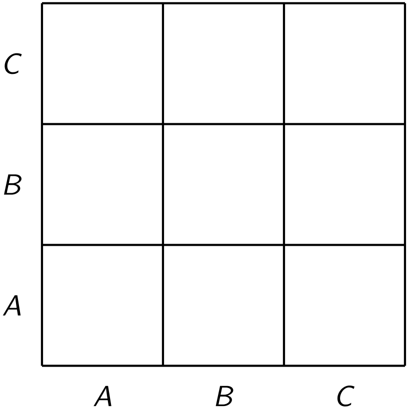
Crucially we have in addition the linear order on the rows and columns $\rightarrow$ monadic independence

Interval graphs are delineated

Right endpoint ordering witnesses bounded twin-width or

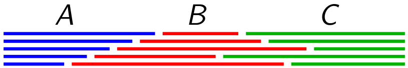


Back to the mixed minors

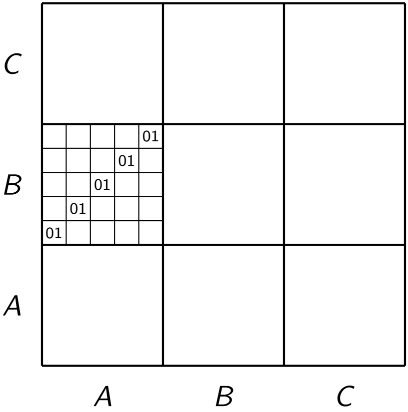


Interval graphs are delineated

Right endpoint ordering witnesses bounded twin-width or

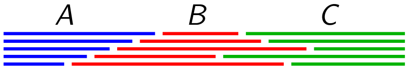


Back to the mixed minors

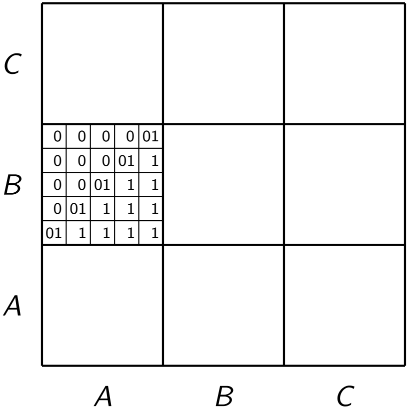


Interval graphs are delineated

Right endpoint ordering witnesses bounded twin-width or

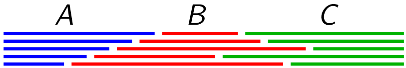


Back to the mixed minors

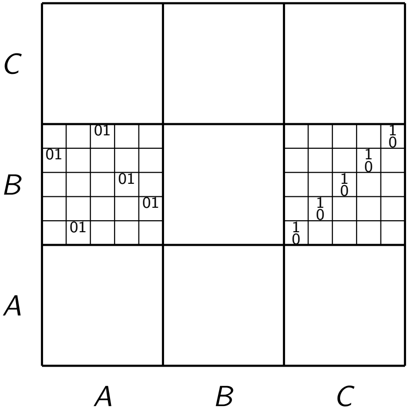


Interval graphs are delineated

Right endpoint ordering witnesses bounded twin-width or



Back to the mixed minors



Non-delineated classes

Exhibit two transdutions T, T' and $\mathcal{C} \subseteq \mathcal{D}$ such that $\mathsf{T}(\mathcal{C})$ contains all subcubic graphs and $\mathsf{T}'(\{\text{subcubic graphs}\})$ contains $\mathcal{C}$

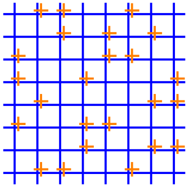
- T implies that $\mathcal{C}$ has unbounded twin-width
- T' implies that $\mathcal{C}$ is monadically dependent

Non-delineated classes

Exhibit two transdutions T, T' and $\mathcal{C} \subseteq \mathcal{D}$ such that $\mathsf{T}(\mathcal{C})$ contains all subcubic graphs and $\mathsf{T}'(\{\text{subcubic graphs}\})$ contains $\mathcal{C}$

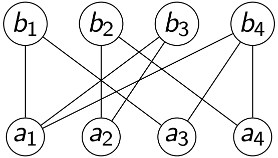
- T implies that $\mathcal{C}$ has unbounded twin-width
- T' implies that $\mathcal{C}$ is monadically dependent

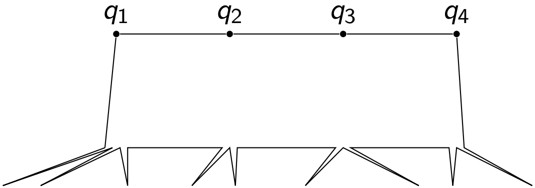
Example: bounded degree, split graphs, **segment graphs**,



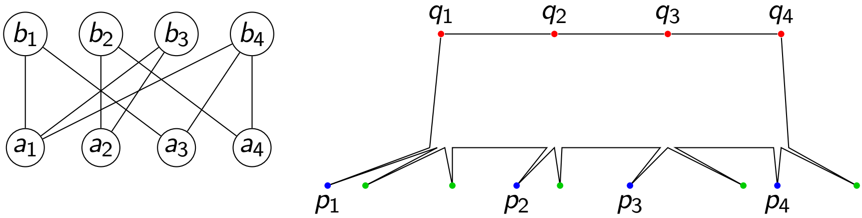
visibility graphs of simple polygons

Simple polygon graphs are not delineated



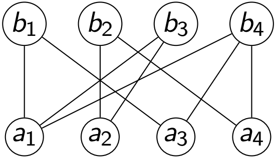


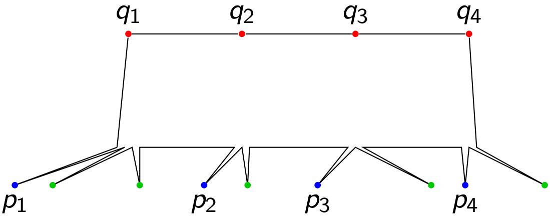
Simple polygon graphs are not delineated



T: *polygons* $\twoheadrightarrow$ *subcubic*

Simple polygon graphs are not delineated

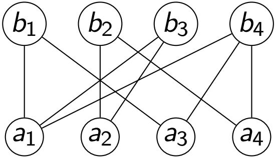


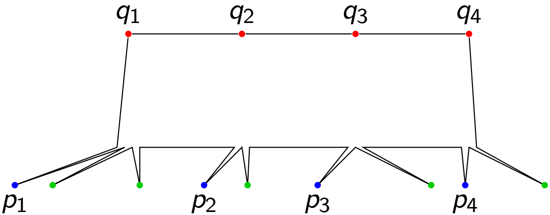


T: *polygons* $\twoheadrightarrow$ *subcubic*

$$\varphi(x, y) = \text{blue}(x) \;\wedge\; \text{red}(y) \;\wedge\; (E(x, y) \vee$$

Simple polygon graphs are not delineated



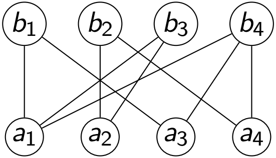


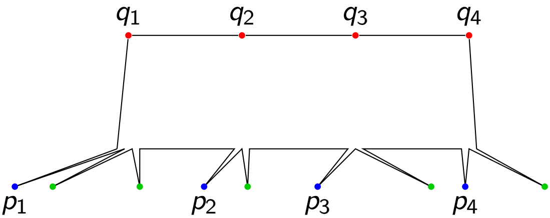
T: *polygons* $\twoheadrightarrow$ *subcubic*

$$\varphi(x, y) = \text{blue}(x) \ \wedge \ \text{red}(y) \ \wedge \ (E(x, y) \vee$$

$$(\exists z_1 \exists z_2 \ \text{black}(z_1) \wedge \text{green}(z_2) \wedge E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, y)) \vee$$

Simple polygon graphs are not delineated





T: *polygons* $\twoheadrightarrow$ *subcubic*

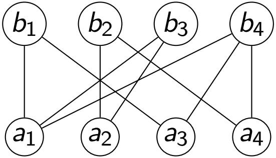
$$\varphi(x,y) = \text{blue}(x) \;\wedge\; \text{red}(y) \;\wedge\; (E(x,y) \vee$$

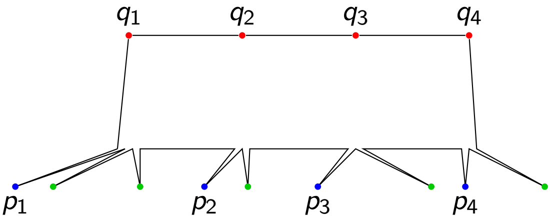
$$(\exists z_1 \exists z_2 \text{ black}(z_1) \wedge \text{green}(z_2) \wedge E(x,z_1) \wedge E(z_1,z_2) \wedge E(z_2,y)) \vee$$

$$(\exists z_1 \exists z_2 \exists z_3 \exists z_4 \text{ black}(z_1) \wedge \text{green}(z_2) \wedge \text{black}(z_3) \wedge \text{green}(z_4)$$

$$\wedge E(x,z_1) \wedge E(z_1,z_2) \wedge E(z_2,z_3) \wedge E(z_3,z_4) \wedge E(z_4,y)))$$

Simple polygon graphs are not delineated





T: *polygons* $\twoheadrightarrow$ *subcubic*

$$\varphi(x, y) = \text{blue}(x) \ \wedge \ \text{red}(y) \ \wedge \ (E(x, y) \vee$$

$$(\exists z_1 \exists z_2 \ \text{black}(z_1) \wedge \text{green}(z_2) \wedge E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, y)) \vee$$

$$(\exists z_1 \exists z_2 \exists z_3 \exists z_4 \ \text{black}(z_1) \wedge \text{green}(z_2) \wedge \text{black}(z_3) \wedge \text{green}(z_4)$$

$$\wedge E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, z_3) \wedge E(z_3, z_4) \wedge E(z_4, y)))$$

T': *subcubic* $\twoheadrightarrow$ *polygons*, add the clique on red and black vertices

Twin-width win-win

Goal: compute FO-definable parameter p in FPT time in $\mathcal{C}$.

Show that $\exists f$ non-decreasing, such that $\forall G \in \mathcal{C}$ an $f(p(G))$-sequence of G can be computed in FPT time

- Width $> f(k)$: report $p(G) > k$
- Width $\leqslant f(k)$: use FO model checking algorithm

Twin-width win-win

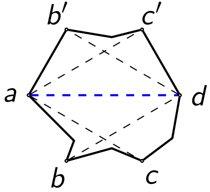
Goal: compute FO-definable parameter p in FPT time in $\mathcal{C}$.

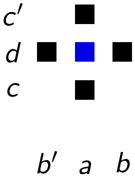
Show that $\exists f$ non-decreasing, such that $\forall G \in \mathcal{C}$ an $f(p(G))$-sequence of G can be computed in FPT time

- Width $> f(k)$: report $p(G) > k$
- Width $\leqslant f(k)$: use FO model checking algorithm

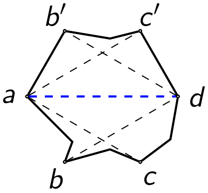
$\rightarrow$ k-Independent Set in visibility graphs of simple polygons

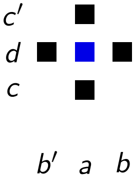
Ordering along the boundary of the polygon

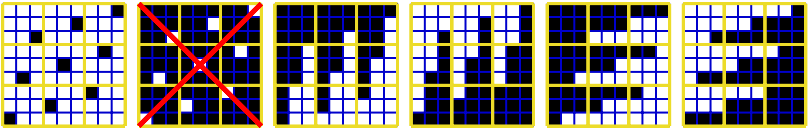




Ordering along the boundary of the polygon

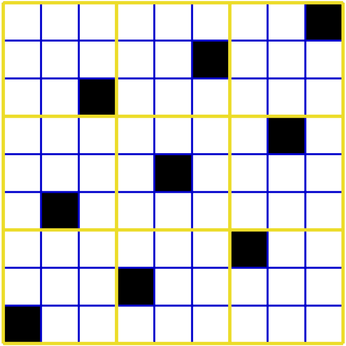


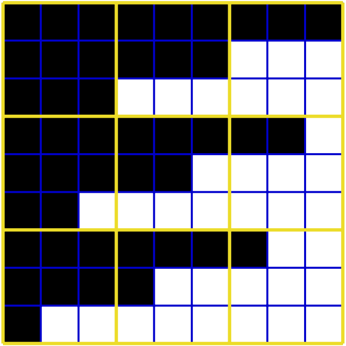




Extractions

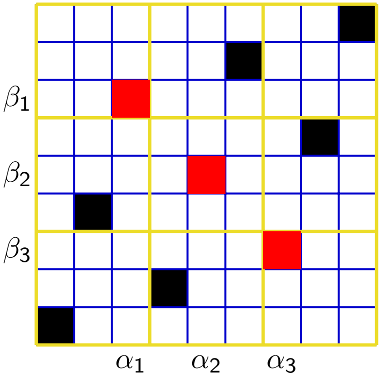
Here we only need a decreasing pattern

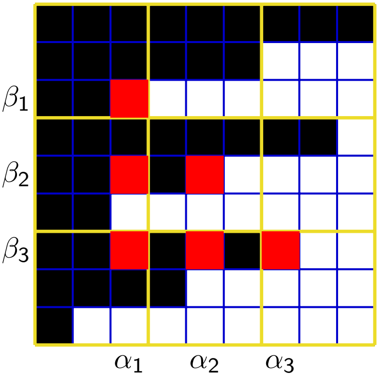




Extractions

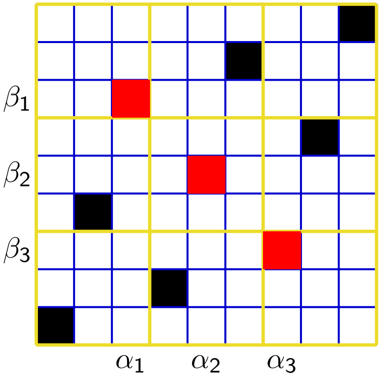
Here we only need a decreasing pattern

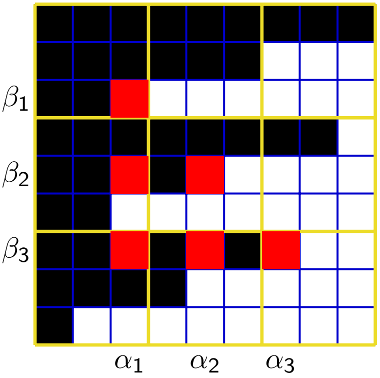




Extractions

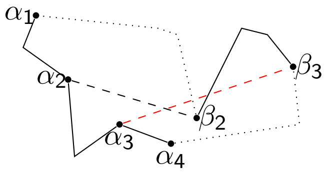
Here we only need a decreasing pattern





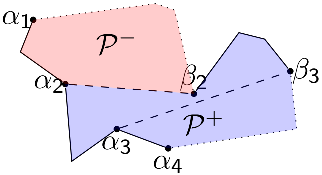
By Ramsey's theorem, we can assume that the α_is and the β_is both induce a clique.

Geometric arguments



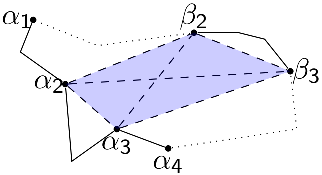
Quadrangle $\alpha_2\alpha_3\beta_3\beta_2$ is not self-crossing

Geometric arguments



Quadrangle $\alpha_2\alpha_3\beta_3\beta_2$ has to be convex

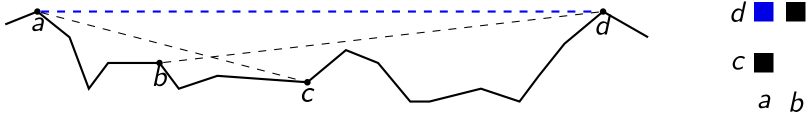
Geometric arguments

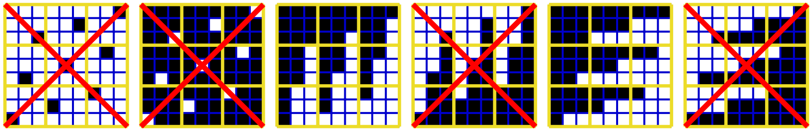


Then $\alpha_2, \alpha_3, \beta_3, \beta_2$ induce K_4, a contradiction

Visibility graphs of 1.5D terrains

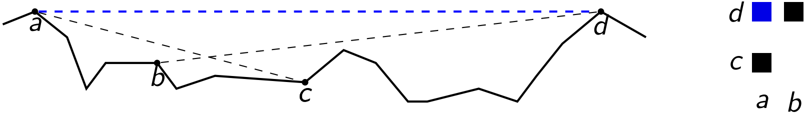
Order along x-coordinates

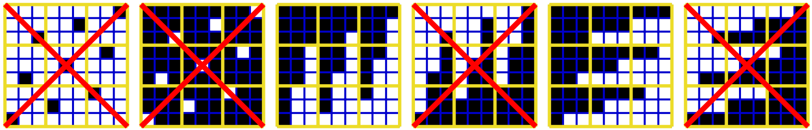




Visibility graphs of 1.5D terrains

Order along x-coordinates





k-Biclique and k-Ladder are FPT in this class

Questions on delineation

Question (Yes! Geniet, Thomassé '22+)

Are tournaments delineated?

Question

Are visibility graphs of terrains delineated?

Question

Are unit segments delineated?

Question

Is non delineation equivalent to having a subclass transduction equivalent to subcubic graphs?

Questions on delineation

Question (Yes! Geniet, Thomassé '22+)

Are tournaments delineated?

Question

Are visibility graphs of terrains delineated?

Question

Are unit segments delineated?

Question

Is non delineation equivalent to having a subclass transduction equivalent to subcubic graphs?

Thank you for your attention!