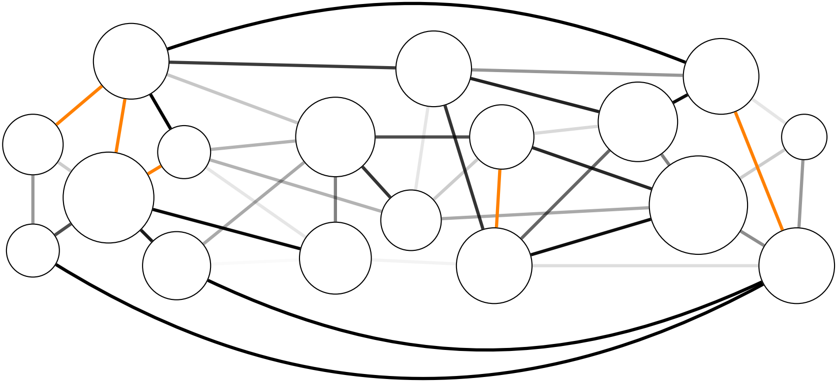
The geometry of twin-width

Édouard Bonnet

ENS Lyon, LIP

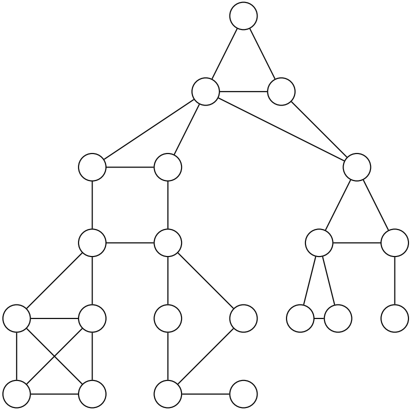
June 30th, 2022, Datashape seminar

Sketching a graph - Szemerédi's Regularity Lemma

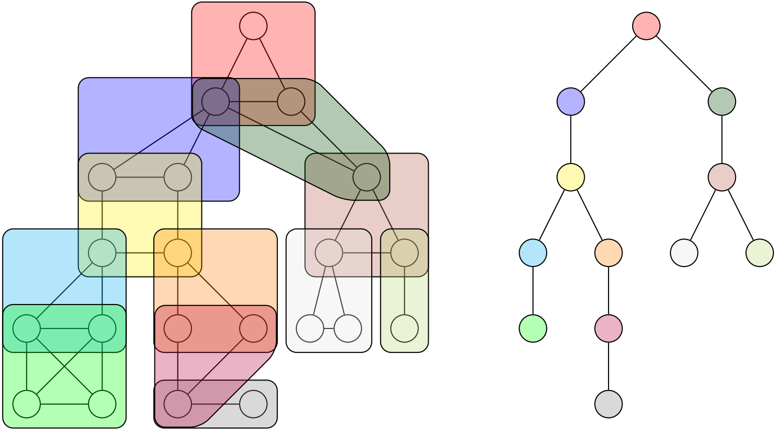


Every graph can be vertex-partitioned into a constant number of (balanced) parts such that there is a random-like edge set between every but an arbitrarily small fraction of pairs of parts.

Sketching a graph - Tree decompositions

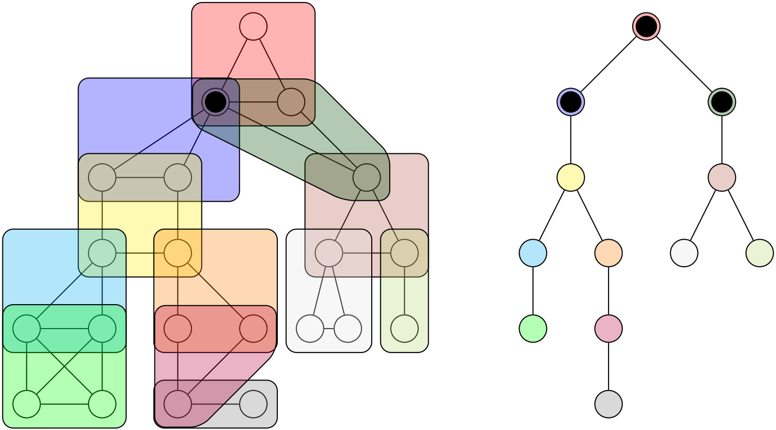


Sketching a graph - Tree decompositions



Edge cover by vertex subsets (called bags) mapping to a tree such that the bags containing any fixed vertex map to a subtree

Sketching a graph - Tree decompositions



Edge cover by vertex subsets (called bags) mapping to a tree such that the bags containing any fixed vertex map to a subtree

Limits

Two immense successes in combinatorics, algorithms, etc. but:

- the former is only meaningful in dense graphs ($m = \Omega(n^2)$)
- the latter is most helpful in sparse graphs ($m = O(n)$)

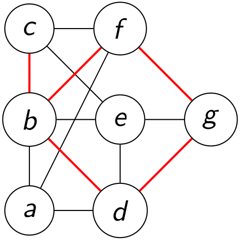
Limits

Two immense successes in combinatorics, algorithms, etc. but:

- the former is only meaningful in dense graphs ($m = \Omega(n^2)$)
- the latter is most helpful in sparse graphs ($m = O(n)$)

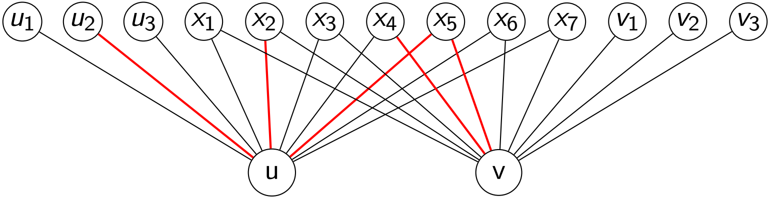
Other useful ways to approximate a graph?

Trigraphs



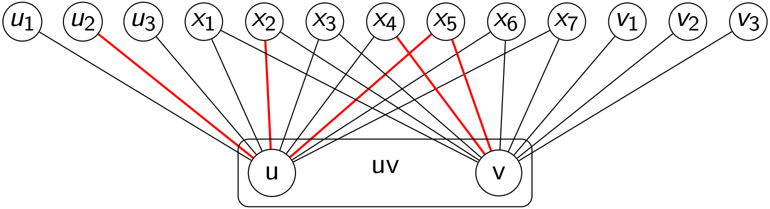
Three outcomes between a pair of vertices:
edge, or non-edge, or red edge (error edge)

Contractions in trigraphs



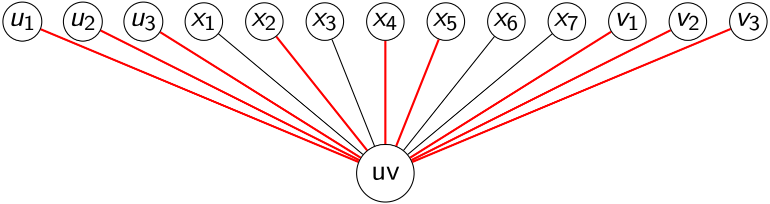
Identification of two non-necessarily adjacent vertices

Contractions in trigraphs



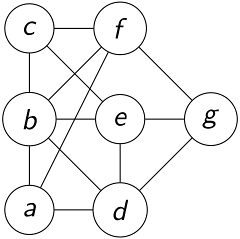
Identification of two non-necessarily adjacent vertices

Contractions in trigraphs



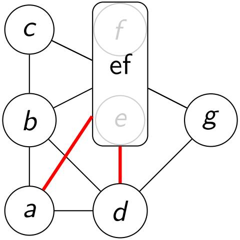
edges to $N(u) \triangle N(v)$ turn red, for $N(u) \cap N(v)$ red is absorbing

Contraction sequence



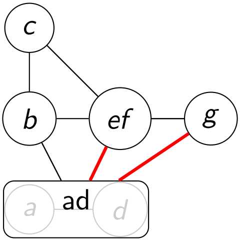
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



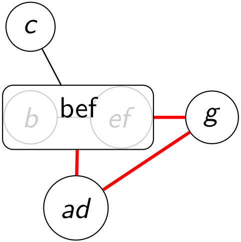
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



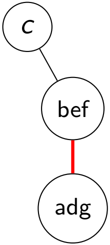
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



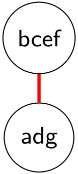
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that
G_i is obtained by performing one contraction in G_{i+1}.

Contraction sequence



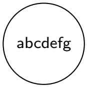
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that
G_i is obtained by performing one contraction in G_{i+1}.

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}.

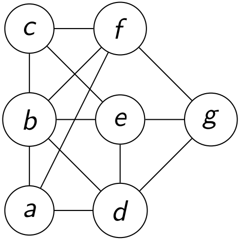
Contraction sequence



A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ 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.

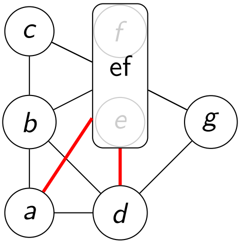


Maximum red degree $= 0$

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.

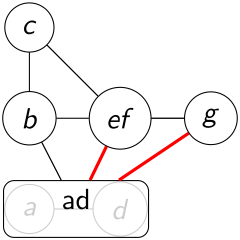


Maximum red degree $= 2$

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.

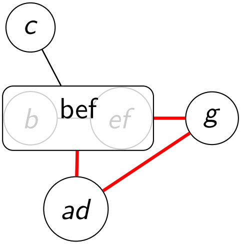


Maximum red degree $= 2$

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.

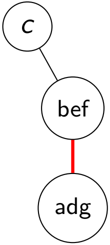


Maximum red degree = 2

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.

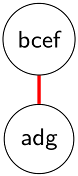


Maximum red degree $= 1$

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.

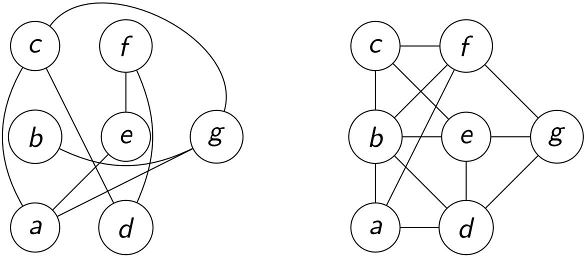
Maximum red degree $= 0$

overall maximum red degree = 2

Simple operations preserving small twin-width

- complementation: remains the same
- taking induced subgraphs: may only decrease
- adding one vertex linked arbitrarily: at most “doubles”
- substitution and lexicographic product

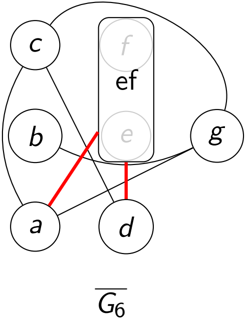
Complementation

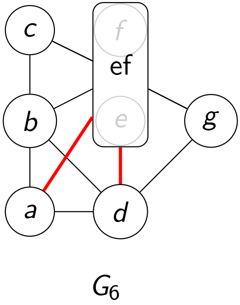


$\overline{G}$ $\quad$ G

$$\text{tww}(\overline{G}) = \text{tww}(G)$$

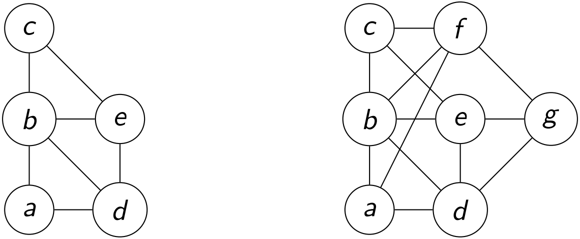
Complementation





$$\text{tww}(\overline{G}) = \text{tww}(G)$$

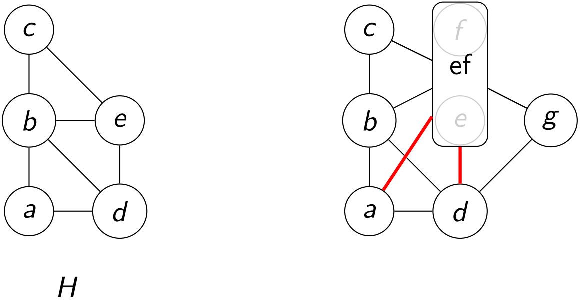
Induced subgraph



H G

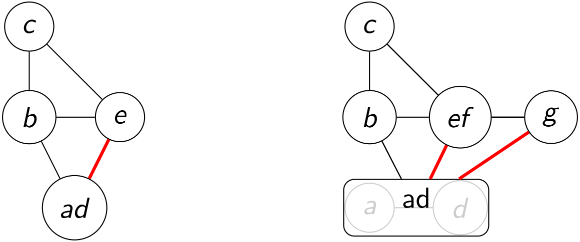
$$\text{tww}(H) \leqslant \text{tww}(G)$$

Induced subgraph



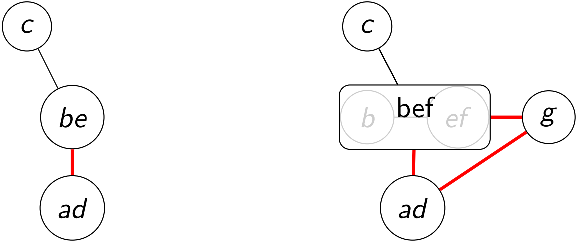
Ignore absent vertices

Induced subgraph



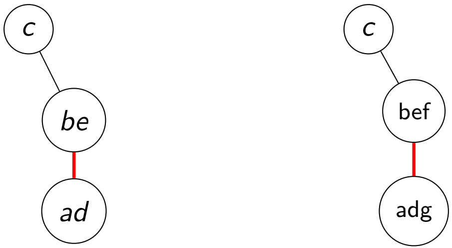
Mimic the contractions otherwise

Induced subgraph



Mimic the contractions otherwise

Induced subgraph



Mimic the contractions otherwise

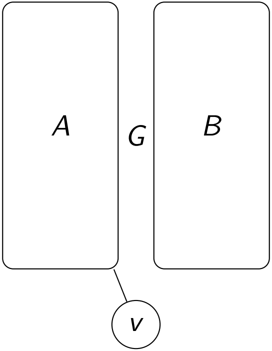
Induced subgraph

Mimic the contractions otherwise

Induced subgraph

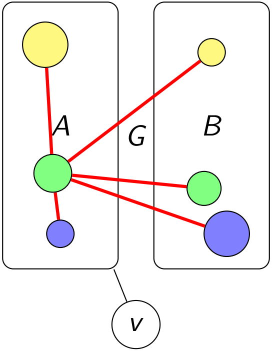
Mimic the contractions otherwise

Adding one vertex v arbitrarily linked



Split every part into their part in A and in B until the very end

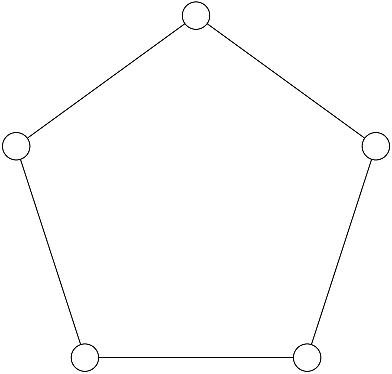
Adding one vertex v arbitrarily linked



Split every part into their part in A and in B until the very end

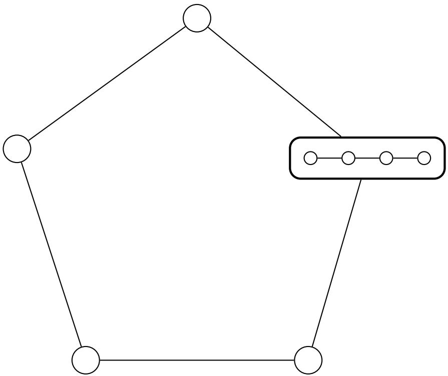
$$\text{tww}(G + v) \leqslant 2 \cdot \text{tww}(G) + 1$$

Substitution and lexicographic product



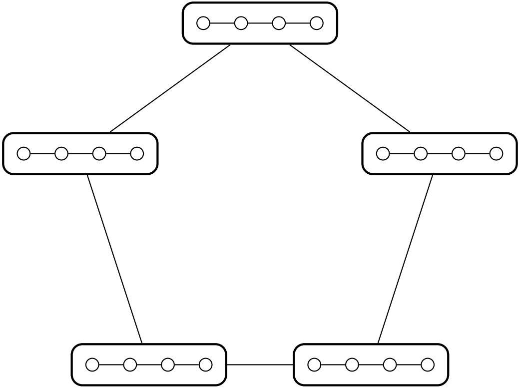
$G = C_5$

Substitution and lexicographic product



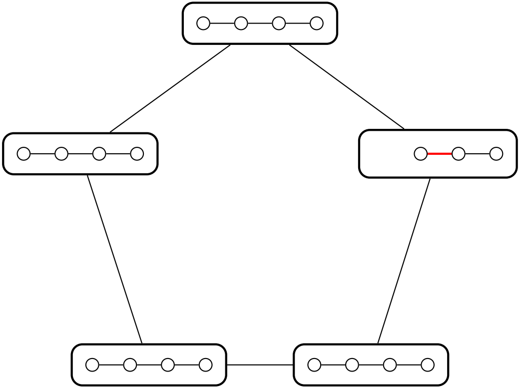
$G = C_5$, $H = P_4$, substitution $G[v \leftarrow H]$

Substitution and lexicographic product



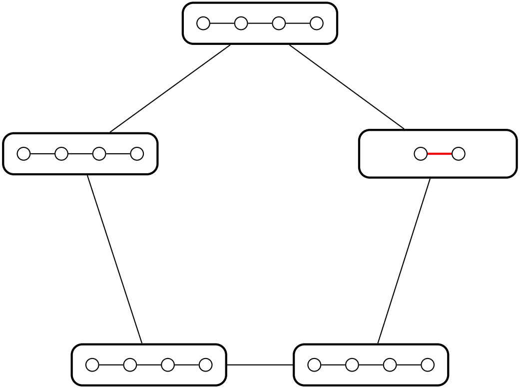
$G = C_5$, $H = P_4$, lexicographic product $G[H]$

Substitution and lexicographic product



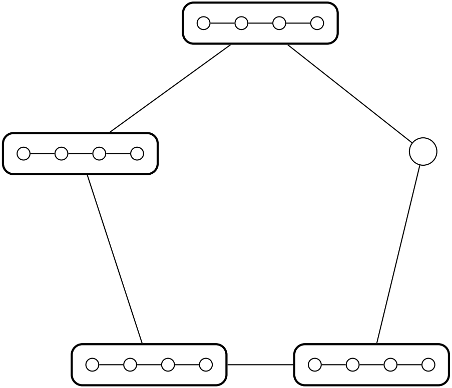
More generally any modular decomposition

Substitution and lexicographic product



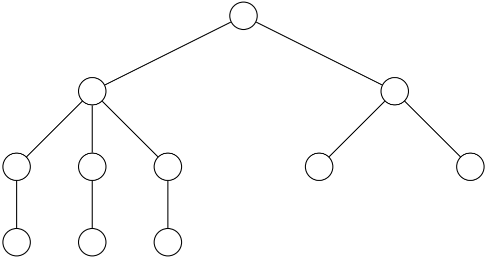
More generally any modular decomposition

Substitution and lexicographic product



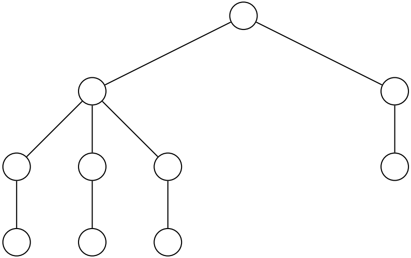
$$\text{tww}(G[H]) = \max(\text{tww}(G), \text{tww}(H))$$

Graphs with bounded twin-width – trees



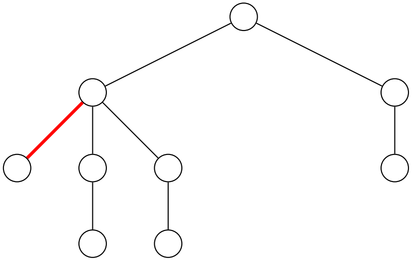
If possible, contract two twin leaves

Graphs with bounded twin-width – trees



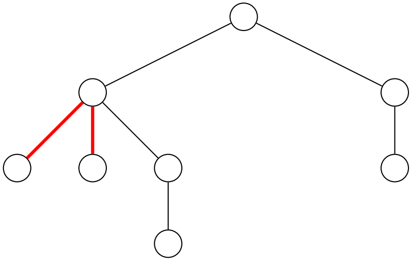
If not, contract a deepest leaf with its parent

Graphs with bounded twin-width – trees



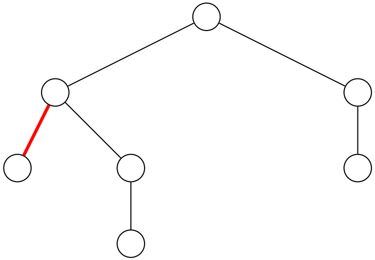
If not, contract a deepest leaf with its parent

Graphs with bounded twin-width – trees



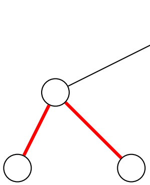
If possible, contract two twin leaves

Graphs with bounded twin-width – trees



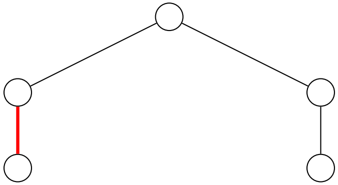
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



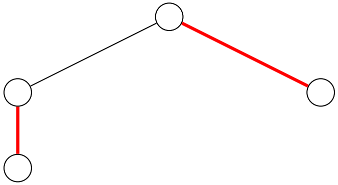
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



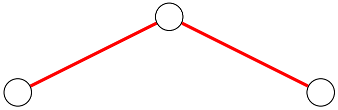
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



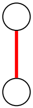
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees

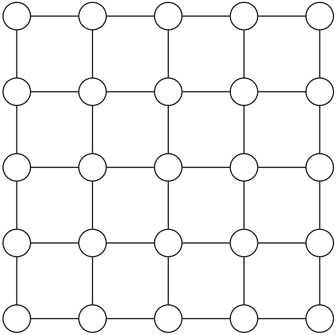


Cannot create a red degree-3 vertex

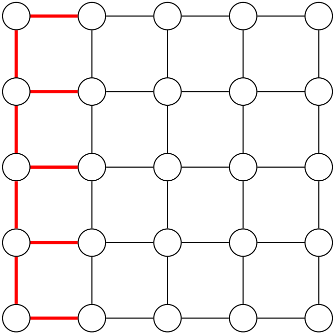
Graphs with bounded twin-width – trees

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

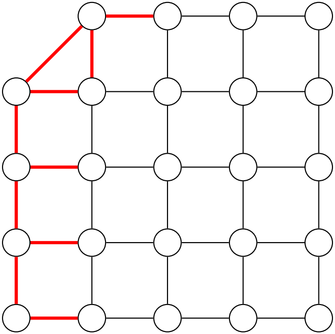
Graphs with bounded twin-width – grids



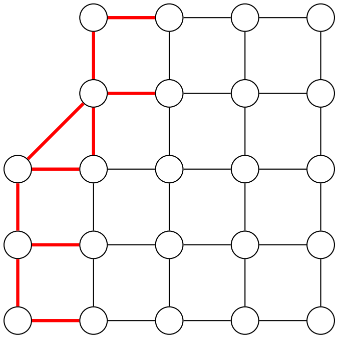
Graphs with bounded twin-width – grids



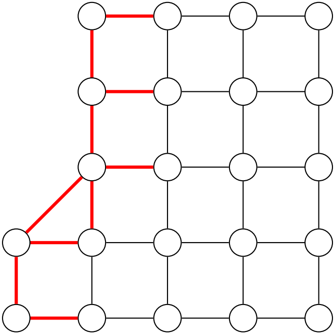
Graphs with bounded twin-width – grids



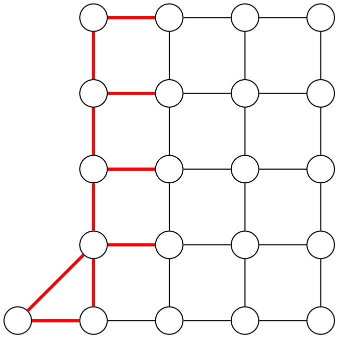
Graphs with bounded twin-width – grids



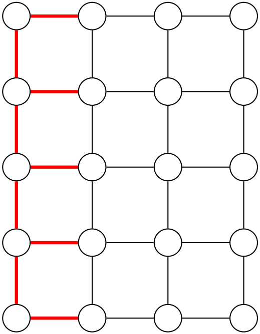
Graphs with bounded twin-width – grids



Graphs with bounded twin-width – grids

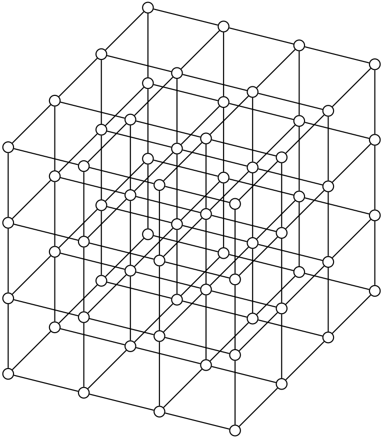


Graphs with bounded twin-width – grids

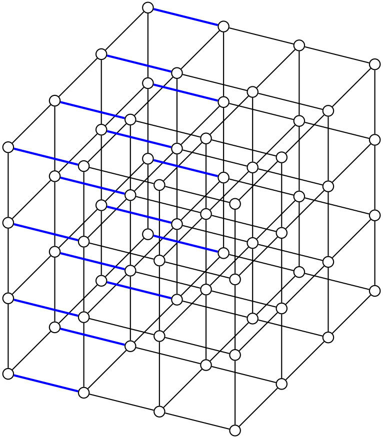


4-sequence for planar grids

3-dimensional grids

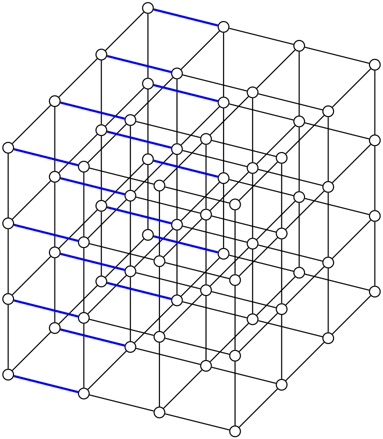


3-dimensional grids



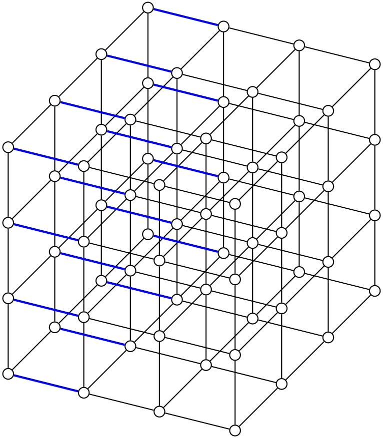
Contract the blue edges in any order $\rightarrow$ 12-sequence

3-dimensional grids



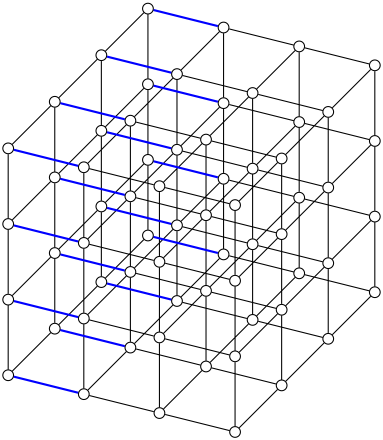
The d-dimensional grid has twin-width $\leqslant 4d$ (even $3d$)

3-dimensional grids



K_t-free unit d-dimensional disk graphs

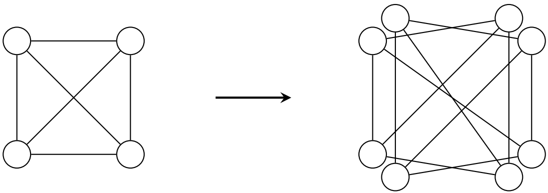
3-dimensional grids



~~K_t-free unit d-dimensional disk graphs~~

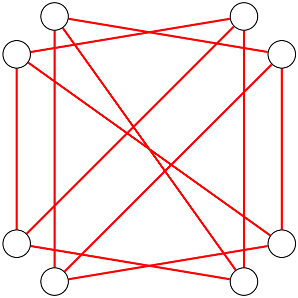
1-skeletons of Rips complexes of point sets in $\mathbb{R}^d$ with dimension less than t have bounded twin-width

2-lifts, expanders with bounded twin-width



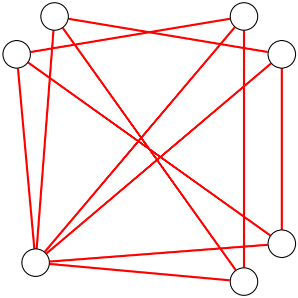
split each vertex in 2, replace each edge by 1 of the 2 matchings

2-lifts, expanders with bounded twin-width



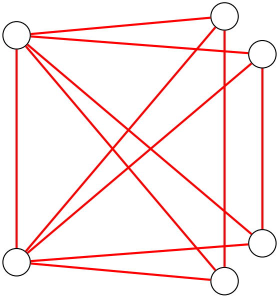
Iterated 2-lifts of K_4 have twin-width at most 6

2-lifts, expanders with bounded twin-width



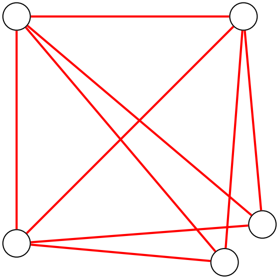
Iterated 2-lifts of K_4 have twin-width at most 6

2-lifts, expanders with bounded twin-width



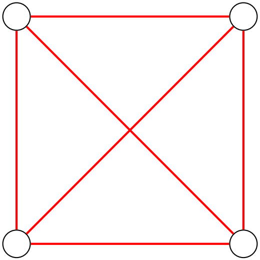
Iterated 2-lifts of K_4 have twin-width at most 6

2-lifts, expanders with bounded twin-width



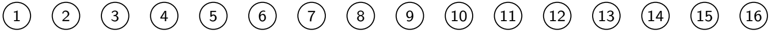
Iterated 2-lifts of K_4 have twin-width at most 6

2-lifts, expanders with bounded twin-width

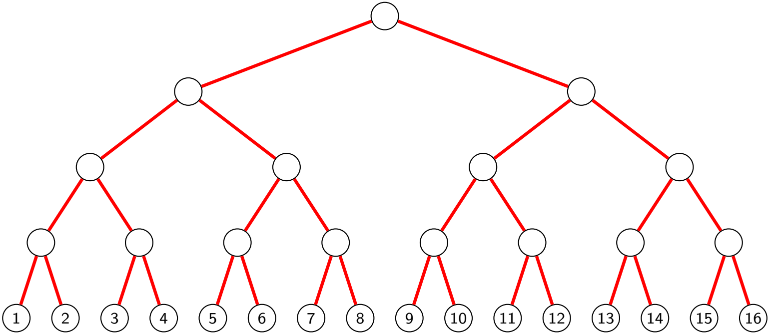


Iterated 2-lifts of K_4 have twin-width at most 6
treewidth $\Omega(n)$

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

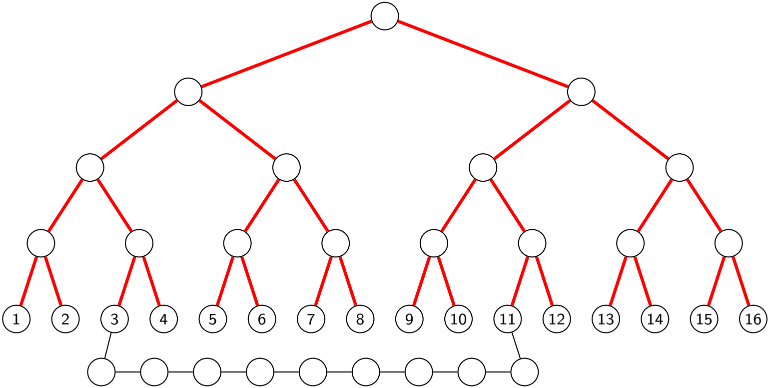


($\geqslant 2 \log n$)-subdivisions have twin-width at most 4



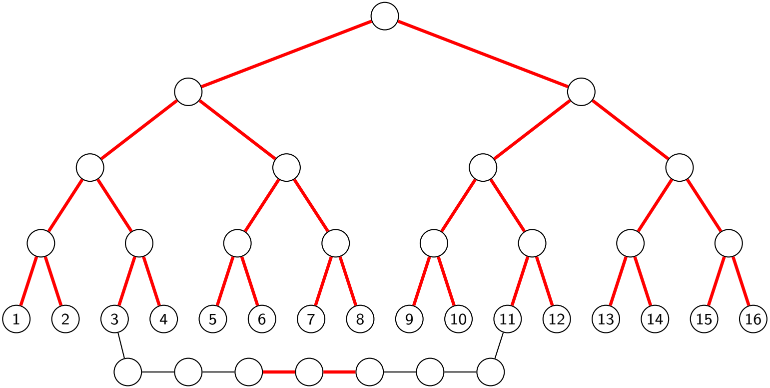
Add a red full binary tree whose leaves are the vertex set

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



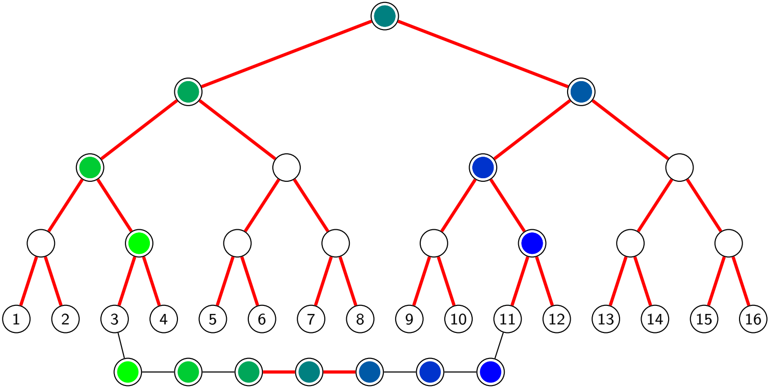
Take any subdivided edge

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



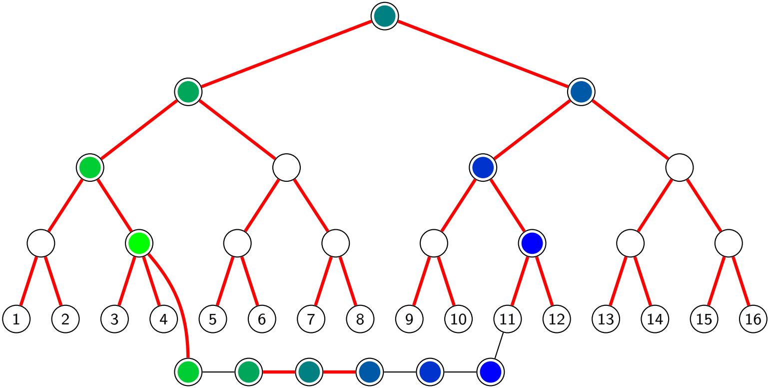
Shorten it to the length of the path in the red tree

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



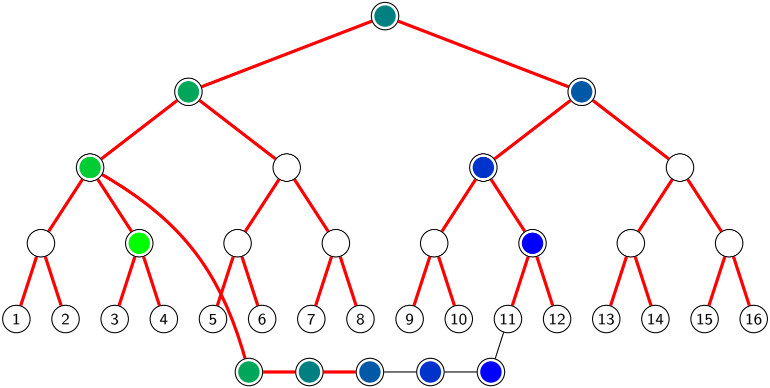
Zip the subdivided edge in the tree

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



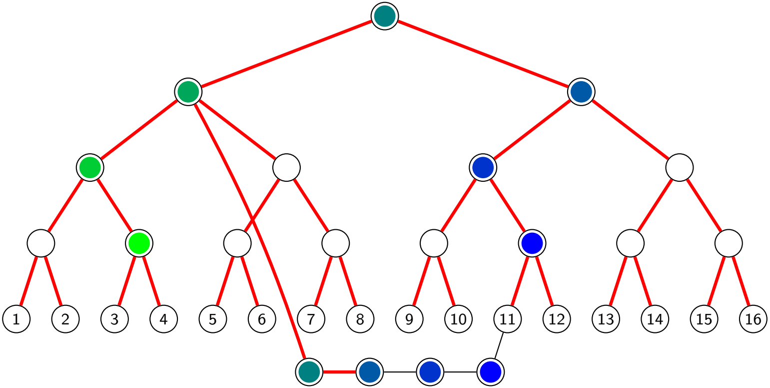
Zip the subdivided edge in the tree

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



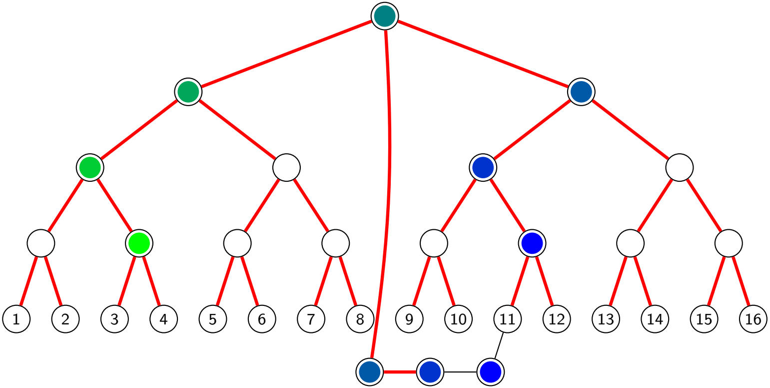
Zip the subdivided edge in the tree

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



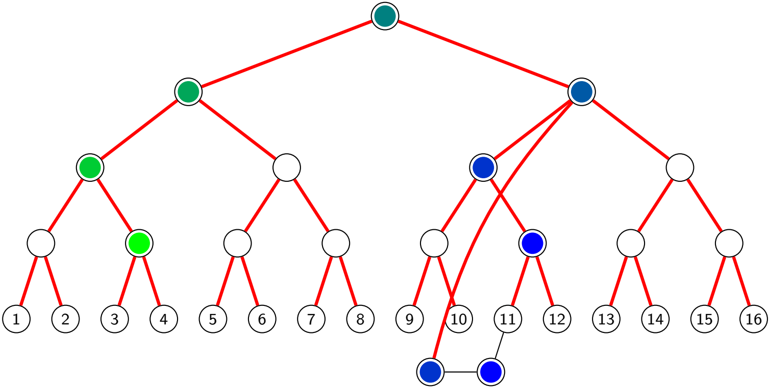
Zip the subdivided edge in the tree

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



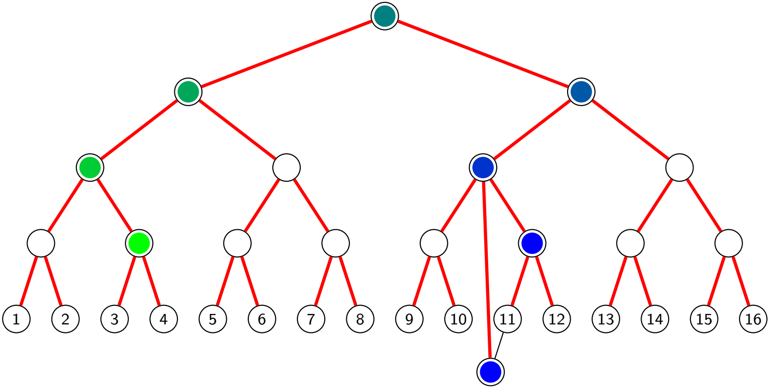
Zip the subdivided edge in the tree

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



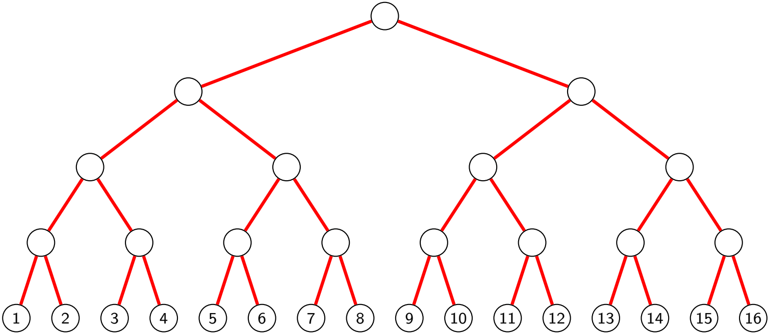
Zip the subdivided edge in the tree

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



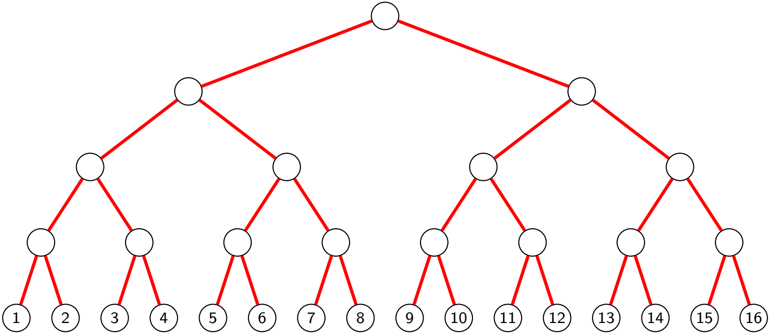
Zip the subdivided edge in the tree

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



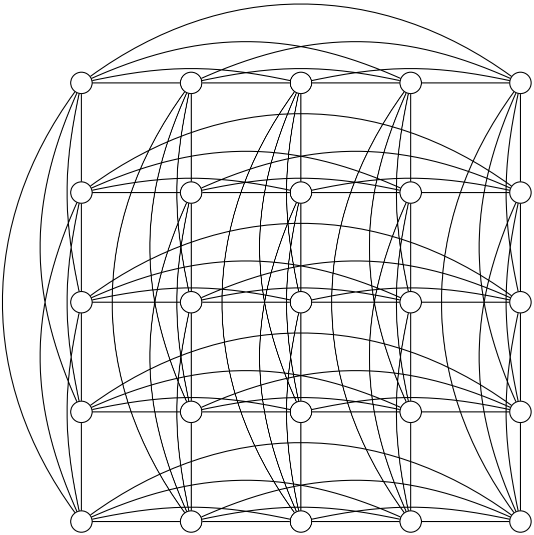
Move to the next subdivided edge also of unbounded cliquewidth

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



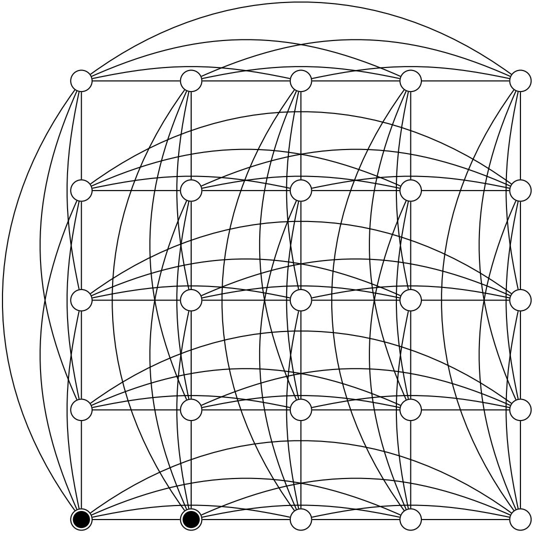
Twin-width is not topological (in the graph-theoretic sense)

First example of unbounded twin-width



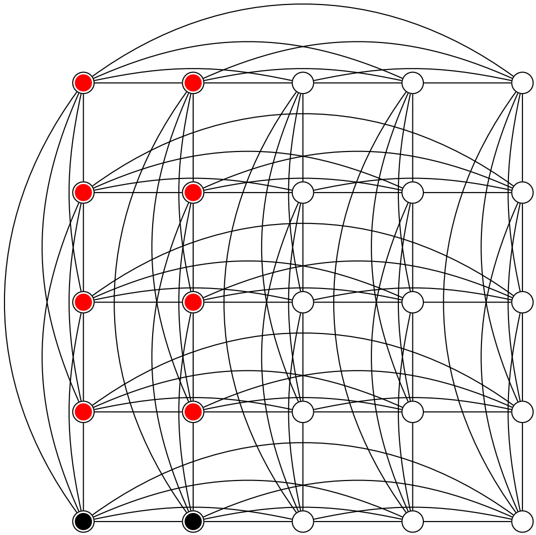
Line graph of a biclique a.k.a. rook graph

First example of unbounded twin-width



No pair of near twins

First example of unbounded twin-width



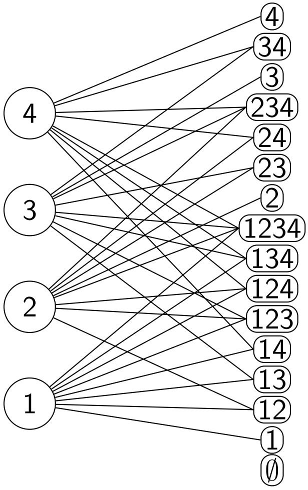
No pair of near twins

Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.

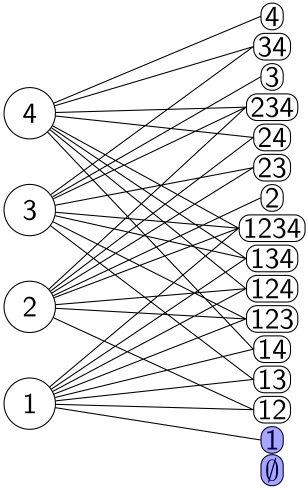
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



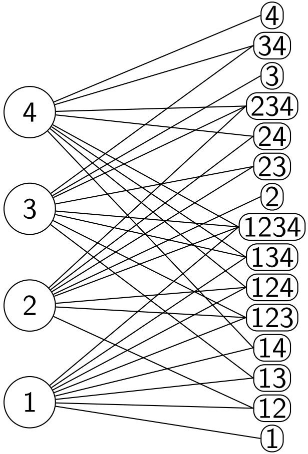
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



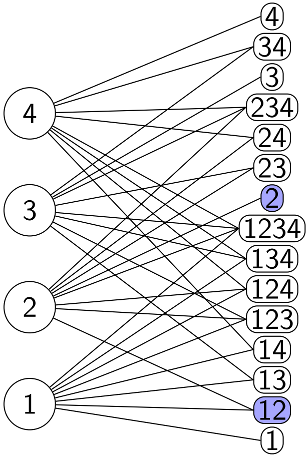
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



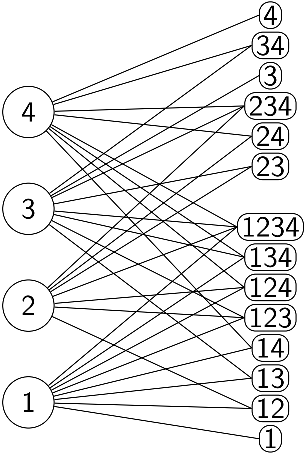
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



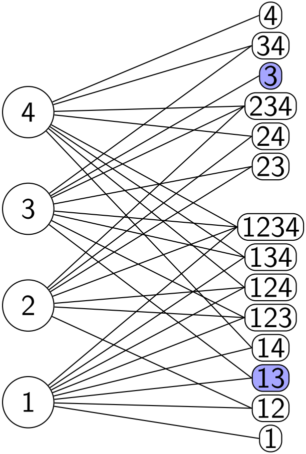
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



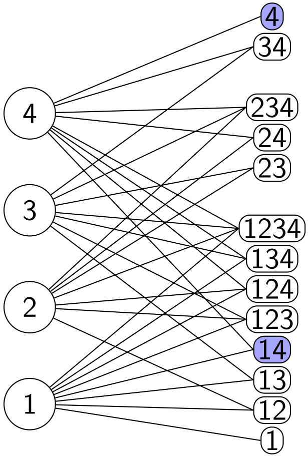
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



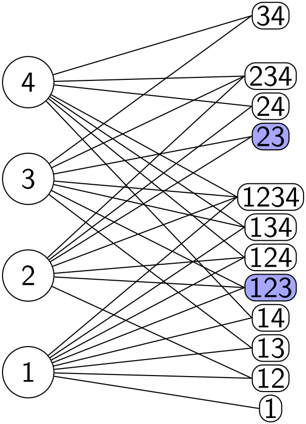
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



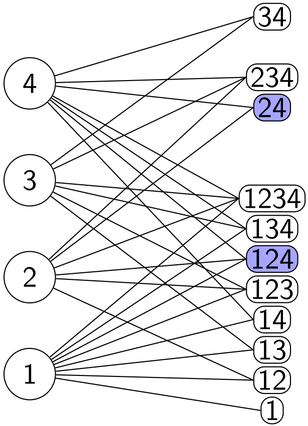
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



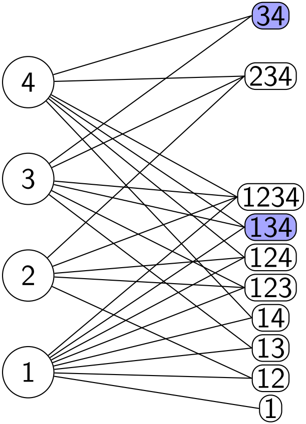
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



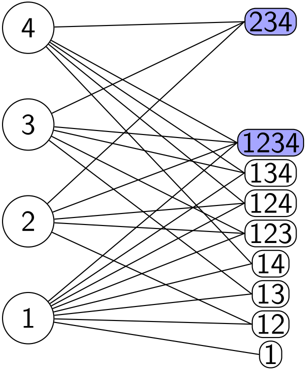
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



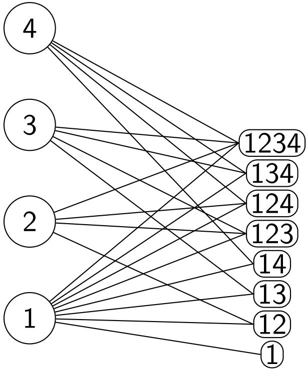
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



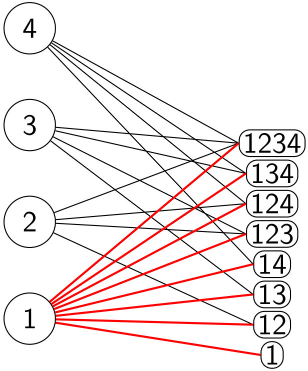
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



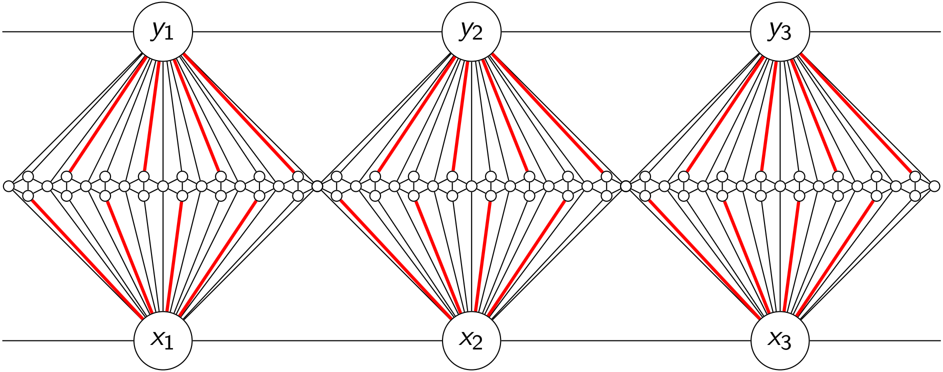
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



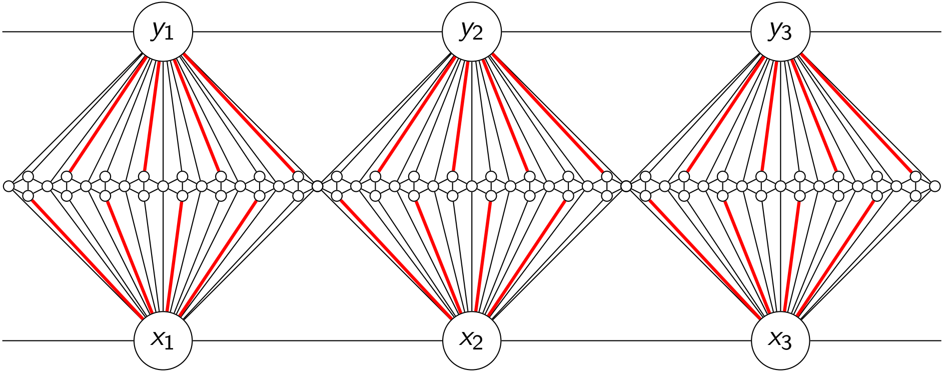
Graphs with bounded twin-width – planar graphs?

Graphs with bounded twin-width – planar graphs?



For every d, a planar trigraph without planar d-contraction

Graphs with bounded twin-width – planar graphs?



For every d, a planar trigraph without planar d-contraction

More powerfool tool needed

Bounded twin-width via adjacency matrices

A matrix is t-**mixed free** if it does not have a $t \times t$ division where every cell has two distinct rows and two distinct columns

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

Bounded twin-width via adjacency matrices

A matrix is t-**mixed free** if it does not have a $t \times t$ division where every cell has two distinct rows and two distinct columns

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

Bounded twin-width via adjacency matrices

A matrix is t-**mixed free** if it does not have a $t \times t$ division where every cell has two distinct rows and two distinct columns

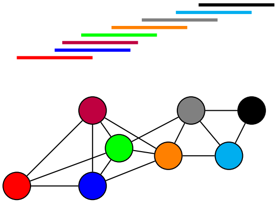
$$\left[\begin{array}{cc|ccc|ccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

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

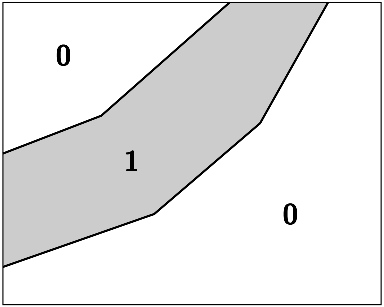
A graph class has bounded twin-width if and only if all its graphs admit an $O(1)$-mixed free adjacency matrix.

Unit interval graphs

Intersection graph of unit segments on the real line

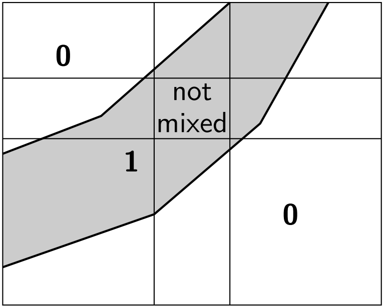


Unit interval graphs



order by left endpoints

Unit interval graphs



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

Graph minors

Formed by **vertex deletion**, **edge deletion**, and **edge contraction**

A graph G is *H-minor free* if H is not a minor of G

A graph class is *H-minor free* if all its graphs are

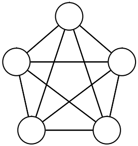
Graph minors

Formed by **vertex deletion**, **edge deletion**, and **edge contraction**

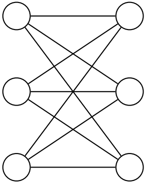
A graph G is *H-minor free* if H is not a minor of G

A graph class is *H-minor free* if all its graphs are

Planar graphs are exactly the graphs without K_5 or $K_{3,3}$ as a minor



K_5

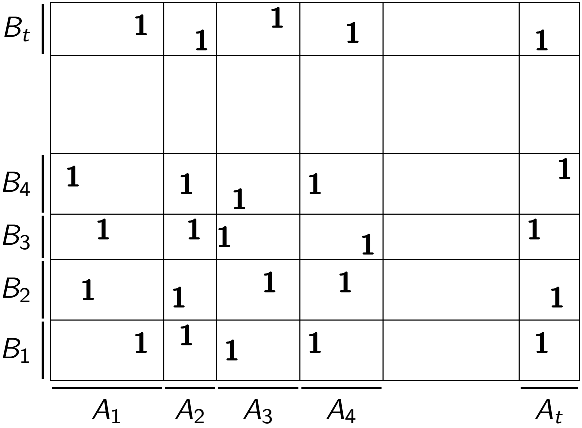


$K_{3,3}$

Bounded twin-width – K_t-minor free graphs

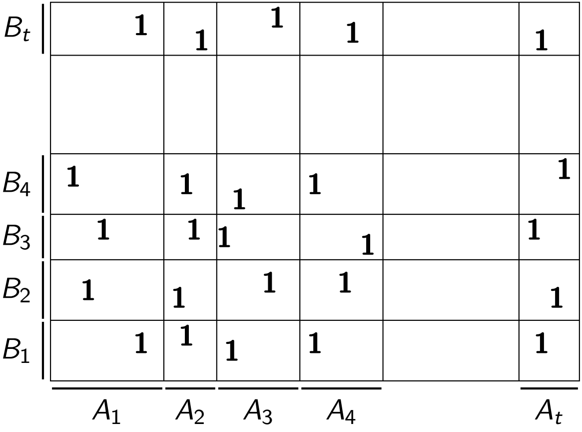
Given a hamiltonian path, we would just use this order

Bounded twin-width – K_t-minor free graphs



Contracting the $2t$ subpaths yields a $K_{t,t}$-minor, hence a K_t-minor

Bounded twin-width – K_t-minor free graphs



Instead we use a specially crafted lex-DFS discovery order

Better upper bounds

Theorem (Hlinený '22)

Planar graphs have twin-width at most 11.

Theorem (B., Kwon, Wood '22)

Graphs of genus g have twin-width $O(g)$.

Better upper bounds

Theorem (Hlinený '22)

Planar graphs have twin-width at most 11.

Theorem (B., Kwon, Wood '22)

Graphs of genus g have twin-width $O(g)$.

Inspired and based on the recent graph product structure theorem:

Theorem (Dujmovic, Joret, Micek, Morin, Ueckerdt, Wood '19)

Every planar graph is the subgraph of the strong product of a path and a graph of treewidth at most 8.

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, and even, boolean-width graphs,*
- *every hereditary proper subclass of permutation graphs,*
- *posets of bounded antichain size (seen as digraphs),*
- *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, and even, boolean-width graphs,*
- *every hereditary proper subclass of permutation graphs,*
- *posets of bounded antichain size (seen as digraphs),*
- *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.*

Is this filtration any useful?

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

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: $G \models \varphi$?	

Example:

$$\varphi = \exists x_1 \exists y_1 \cdots \exists x_k \exists y_k \bigwedge_{\{x,y\} \in \binom{\{x_1, y_1, \ldots, x_k, y_k\}}{2}} x \neq y$$

$$\wedge\ E(x, y) \Leftrightarrow \bigvee_{1 \leqslant i \leqslant k} (x = x_i \wedge y = y_i) \vee (x = y_i \wedge y = x_i)$$

$G \models \varphi? \Leftrightarrow$

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: $G \models \varphi$?	

Example:

$$\varphi = \exists x_1 \exists y_1 \cdots \exists x_k \exists y_k \bigwedge_{\{x,y\} \in \binom{\{x_1, y_1, \ldots, x_k, y_k\}}{2}} x \neq y$$

$$\wedge\ E(x,y) \Leftrightarrow \bigvee_{1 \leqslant i \leqslant k} (x = x_i \wedge y = y_i) \vee (x = y_i \wedge y = x_i)$$

$G \models \varphi$? $\Leftrightarrow$ k-INDUCED MATCHING

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: $G \models \varphi$?	

Example:

$$\varphi = \bigvee_{1 \leqslant q \leqslant k,\ q \text{ is odd}} \exists x_1 \notin \{s\}\ E(s, x_1) \wedge (\forall x_2 \notin \{s, x_1\}\ \neg E(x_1, x_2) \vee$$

$$(\exists x_3 \notin \{s, x_1, x_2\}\ E(x_2, x_3) \wedge (\forall x_4 \cdots (\exists x_q \notin \{s, x_1, \ldots, x_{q-1}\} E(x_{q-1}, x_q)$$

$$\wedge\ (\forall x_{q+1}\ \neg E(x_q, x_{q+1}) \vee x_{q+1} \in \{s, x_1, \ldots, x_q\})) \cdots)))$$

$G \models \varphi? \Leftrightarrow$

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: $G \models \varphi$?	

Example:

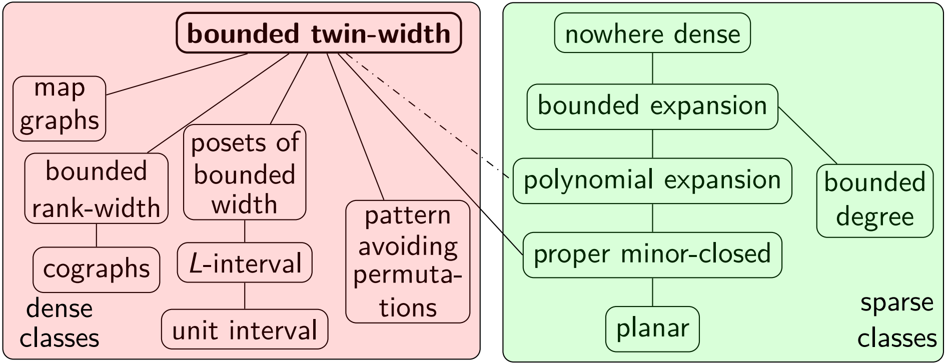
$$\varphi = \bigvee_{1 \leqslant q \leqslant k,\ q \text{ is odd}} \exists x_1 \notin \{s\}\ E(s, x_1) \wedge (\forall x_2 \notin \{s, x_1\}\ \neg E(x_1, x_2) \vee$$

$$(\exists x_3 \notin \{s, x_1, x_2\}\ E(x_2, x_3) \wedge (\forall x_4 \cdots (\exists x_q \notin \{s, x_1, \ldots, x_{q-1}\} E(x_{q-1}, x_q)$$

$$\wedge\ (\forall x_{q+1}\ \neg E(x_q, x_{q+1}) \vee x_{q+1} \in \{s, x_1, \ldots, x_q\})) \cdots)))$$

$G \models \varphi$? $\Leftrightarrow$ SHORT GENERALIZED GEOGRAPHY

Classes with known tractable FO model checking



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

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

FO interpretations and transductions

FO simple 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)

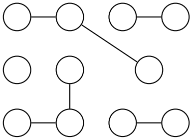
FO interpretations and transductions

FO simple 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



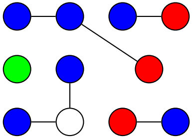
FO interpretations and transductions

FO simple 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



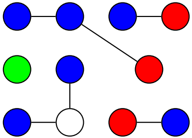
FO interpretations and transductions

FO simple 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



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

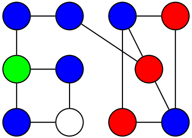
FO interpretations and transductions

FO simple 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

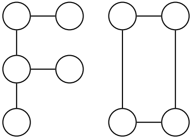


$$\varphi(x,y) = E(x,y) \vee (G(x) \wedge B(y) \wedge \neg\exists z R(z) \wedge E(y,z))$$
$$\vee(R(x) \wedge B(y) \wedge \exists z R(z) \wedge E(y,z) \wedge \neg\exists z B(z) \wedge E(y,z))$$

FO interpretations and transductions

FO simple 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)

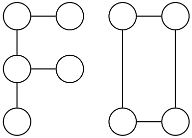
FO transduction: color by $O(1)$ unary relations, interpret, delete



FO interpretations and transductions

FO simple 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)

Transductions of bounded twin-width classes have bounded twin-width.

Small classes

Small: class with at most $n!c^n$ labeled graphs on $[n]$.

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

Bounded twin-width classes are small.

Unifies and extends the same result for:
σ-free permutations [Marcus, Tardos '04]
K_t-minor free graphs [Norine, Seymour, Thomas, Wollan '06]

Small classes

Small: class with at most $n!c^n$ labeled graphs on $[n]$.

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

Bounded twin-width classes are small.

Is the converse true for hereditary classes?

Conjecture (small conjecture)

A hereditary class has bounded twin-width if and only if it is small.

Small classes

Small: class with at most $n!c^n$ labeled graphs on $[n]$.

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

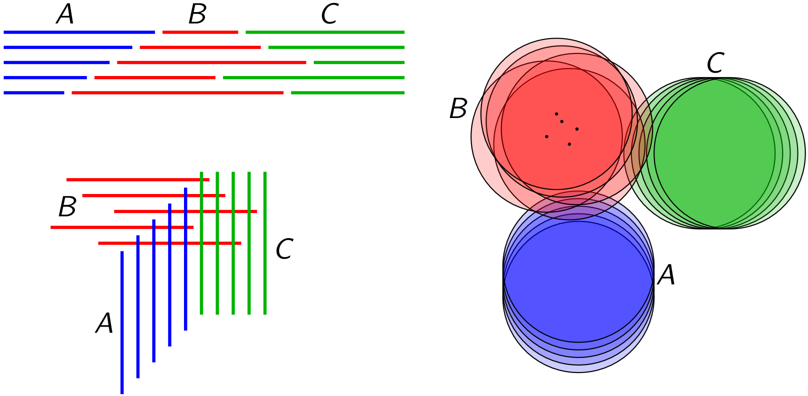
Bounded twin-width classes are small.

Is the converse true for hereditary classes?

Conjecture (refuted: B., Geniet, Tessera, Thomassé '22)

A hereditary class has bounded twin-width if and only if it is small.

Interval graphs, unit disk graphs, triangle-free unit segment graphs have unbounded twin-width



since there are too many such graphs

The transduction tool

Our stability

- adding noise keeps twin-width tamed
- robust notion for finite model theory
- indeed bounded twin-width is the right limit for tractability of FO model checking in ordered graphs (Twin-width IV)
- concrete example: biclique-free segment graphs have bounded twin-width by FO transducing from planar graphs

Reduced parameters

A graph class has bounded reduced X if all its members admit a contraction sequence whose red graphs have bounded X

Reduced parameters

A graph class has bounded reduced X if all its members admit a contraction sequence whose red graphs have bounded X

red graphs have bounded ...	characterize bounded ...
outdegree	(oriented) twin-width
degree	**twin-width**
degree + treewidth	reduced (degree + treewidth)
cutwidth	reduced cutwidth
bandwidth	reduced bandwidth
bandwidth with fixed order	stretch-width
component size	**cliquewidth** (sparse: treewidth)
number of edges*	linear cliquewidth (sparse: pathwidth)

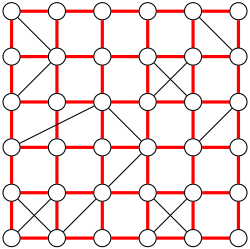
Reduced parameters

A graph class has bounded reduced X if all its members admit a contraction sequence whose red graphs have bounded X

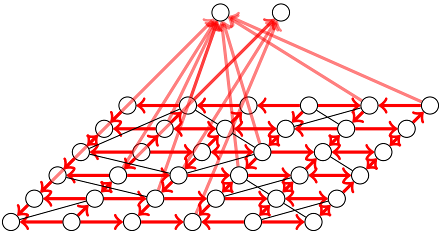
red graphs have bounded ...	characterize bounded ...
outdegree	(oriented) twin-width
degree	**twin-width**
degree + treewidth	reduced (degree + treewidth)
cutwidth	reduced cutwidth
bandwidth	reduced bandwidth
bandwidth with fixed order	stretch-width
component size	**cliquewidth** (sparse: treewidth)
number of edges*	linear cliquewidth (sparse: pathwidth)

Long subdivisions of general graphs have unbounded stretch-width

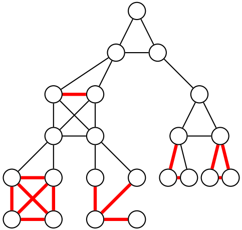
Different conditions imposed on the red graphs



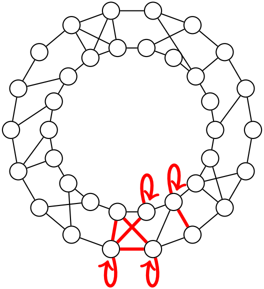
bd degree: defines bd twin-width



bd outdegree: defines bd oriented twin-width



bd component: redefines bd cliquewidth



bd #edges: redefines bd linear cliquewidth

Some directions and questions

- Twin-width of higher arity structures? Simplicial complexes?
- Do compact 3-manifolds have uniformly bounded twin-width?
  (shamelessly borrowed from Kristóf's research project)
- Draw uniformly at random n points in $[0,1]^d$ (or from a different distribution), grow balls from radius 0 to $\sqrt{d}/2$, and track the largest twin-width of the intersection graph. Asymptotics of this function of n?
- Bound the twin-width of some geometric graph classes when half-graphs are forbidden.