A comparison of graph width parameters

Nick Brettell

Victoria University of Wellington, NZ

Joint work with Andrea Munaro,
Daniel Paulusma, and Shizhou Yang

45ACC
12 December 2023

Introduction

A width parameter associates a measure of "width" to a graph.

Width parameters of interest here:

- treewidth, clique-width, mim-width, sim-width, and tree-independence number

- What are these parameters? Why are they of interest?
- How do they relate to each other?

Comparing width parameters

A class of graphs $\mathcal{G}$ has bounded p-width if there exists a constant c such that $p(G) \leq c$ for all $G \in \mathcal{G}$.

For parameters p and q, say p is less restrictive than q if there exists a function f such that $p(G) \leq f(q(G))$ for every graph G.

Then "$\mathcal{G}$ has bounded q-width" $\Rightarrow$ "$\mathcal{G}$ has bounded p-width".

treewidth is the most restrictive parameter we'll consider

For parameters p and q, say p is equivalent to q if p is less restrictive than q, and q is less restrictive than p.

Comparing width parameters

A class of graphs $\mathcal{G}$ has bounded p-width if there exists a constant c such that $p(G) \leq c$ for all $G \in \mathcal{G}$.

For parameters p and q, say p is less restrictive than q if there exists a function f such that $p(G) \leq f(q(G))$ for every graph G.

Then "$\mathcal{G}$ has bounded q-width" $\Rightarrow$ "$\mathcal{G}$ has bounded p-width".

treewidth is the most restrictive parameter we'll consider

For parameters p and q, say p is equivalent to q if p is less restrictive than q, and q is less restrictive than p.

We can also compare parameters on subclasses.

Treewidth

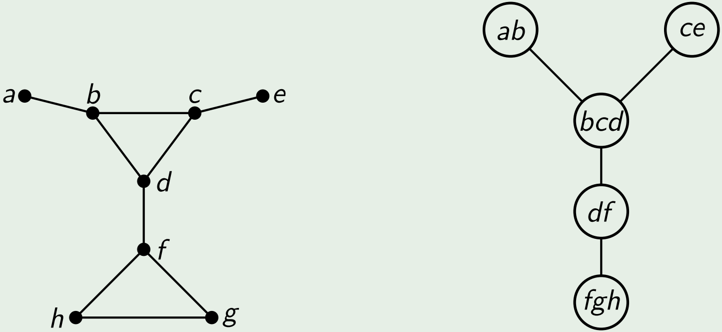
treewidth formalises a notion of how "tree-like" a graph is.

For a graph G, a tree decomposition of G is a tree T and a collection of bags $(B_t)_{t \in V(T)}$ where each bag B_t is a subset of $V(G)$ such that:

1. each vertex of G is in some bag
2. every edge uv of G has both u and v in some bag
3. for each vertex v of G, the bags containing v form a connected subtree of T

The width of a tree decomposition is $\max_{t \in V(T)} |B_t| - 1$.
The treewidth of a graph G, denote $\mathrm{tw}(G)$, is the minimum width among all tree decompositions of G.

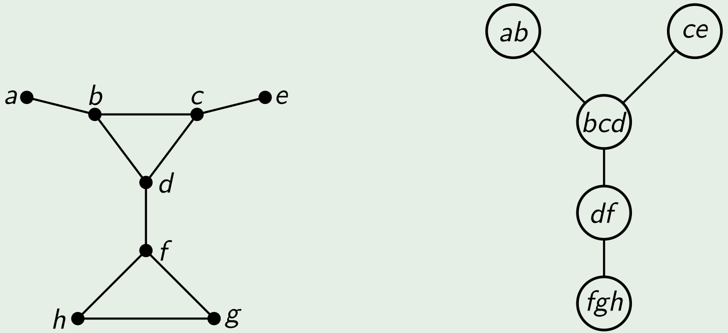
Treewidth: an example



This decomposition has width 2.

$$\mathrm{tw}(G) \leq 2$$

Treewidth: an example



This decomposition has width 2.

$$\mathrm{tw}(G) = 2$$

Algorithms parameterised by treewidth

Many graph problems that are NP-hard in general are polynomial-time solvable for graphs with bounded treewidth.

Theorem (Courcelle, 1990)

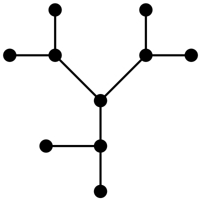
Any graph problem expressible in MSO_2 *logic of graphs is FPT parameterised by treewidth*

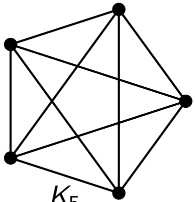
(there is an $f(w) \cdot O(n)$*-time algorithm).*

e.g. Independent Set, k-colouring, ...

Treewidth examples

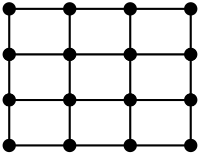
a tree has treewidth 1,





K_5

whereas a complete graph K_{n+1} has treewidth n

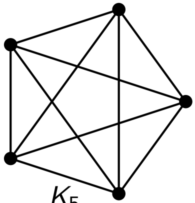


4×4 grid

and an $n \times n$ grid has treewidth n

Treewidth examples

a tree has treewidth 1,



K_5

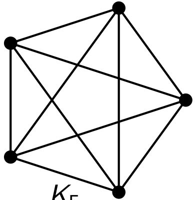
whereas a complete graph K_{n+1} has treewidth n

Bounded treewidth restricts us to relatively sparse graphs.

High treewidth graphs may not have high complexity.

Treewidth examples

a tree has treewidth 1,



K_5

whereas a complete graph K_{n+1} has treewidth n

Bounded treewidth restricts us to relatively sparse graphs.

High treewidth graphs may not have high complexity.

clique-width gives a measure of how "uniformly sparse or dense" a graph is.

Clique-width

a cograph ("complement-reducible graph") can be constructed from K_1's by disjoint unions and joins.

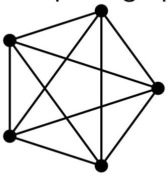
the clique-width of a graph G is the minimum number of labels required to build G by

1. creating a new vertex v with label i
2. disjoint union of two labelled graphs
3. joining by an edge every vertex labeled i to every vertex labeled j
4. relabelling i to j

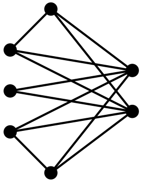
Complementation: $\mathrm{cw}(\overline{G}) \leq 2\mathrm{cw}(G)$

Clique-width examples

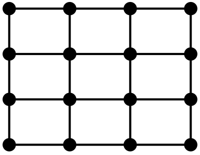
complete graphs have clique-width 1,



cographs have clique-width at most 2,



an $n \times n$ grid has clique-width $n + 1$



Algorithms parameterised by clique-width

Theorem (Courcelle, Makowsky, and Rotics, 2000)

Any problem expressible in MSO_1 *logic of graphs is FPT parameterised by clique-width*

(there is an $f(w) \cdot O(n^3)$*-time algorithm).*

in MSO_1, can't quantify over edge sets
e.g. HAMILTONIAN CYCLE in MSO_2 but not MSO_1

More general class of graphs, less general family of problems

Comparing treewidth and clique-width

Clique-width is less restrictive than treewidth.

[Courcelle and Olariu 2000]
[Corneil and Rotics 2005]

$\mathrm{cw}(G) \leq 3 \cdot 2^{\mathrm{tw}(G)-1}$

However,

Theorem (Gurski and Wanke, 2000)

For every $t \geq 2$, when restricted to the class of graphs with no $K_{t,t}$-subgraph, clique-width is equivalent to treewidth.

Comparing treewidth and clique-width

Clique-width is less restrictive than treewidth.

[Courcelle and Olariu 2000]
[Corneil and Rotics 2005]

$\mathrm{cw}(G) \leq 3 \cdot 2^{\mathrm{tw}(G)-1}$

However,

Theorem (Gurski and Wanke, 2000)

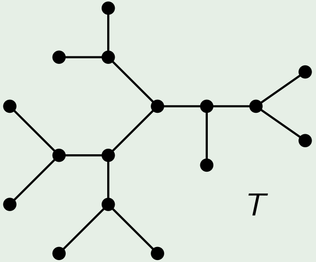
For every $t \geq 2$, when restricted to the class of graphs with no $K_{t,t}$-subgraph, clique-width is equivalent to treewidth.

Theorem (Gurski and Wanke, 2007)

A class of graph $\mathcal{G}$ has bounded treewidth if and only if the class $L(\mathcal{G})$ of line graphs of graphs in $\mathcal{G}$ has bounded clique-width.

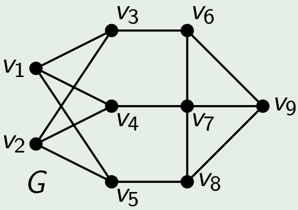
Branch decompositions and f-width

A branch decomposition (T, δ) of G is a subcubic tree T, with a bijection between the leaves of T and $V(G)$.



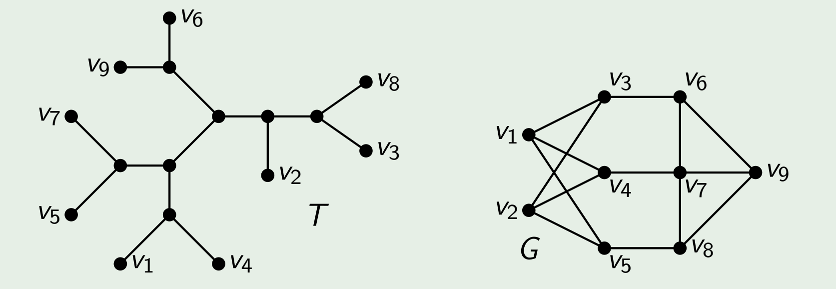
Branch decompositions and f-width

A branch decomposition (T, δ) of G is a subcubic tree T, with a bijection between the leaves of T and $V(G)$.



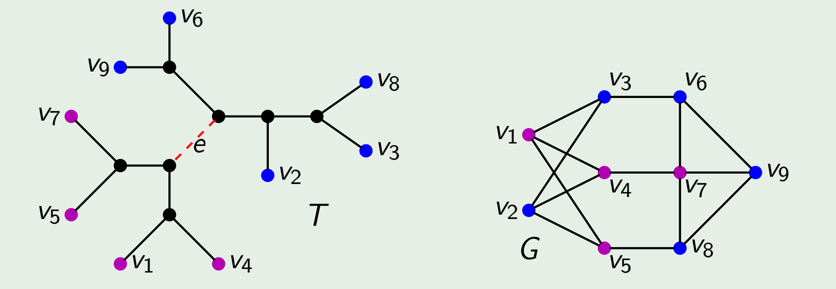
Branch decompositions and f-width

A branch decomposition (T, δ) of G is a subcubic tree T, with a bijection between the leaves of T and $V(G)$.



Branch decompositions and f-width

A branch decomposition (T, δ) of G is a subcubic tree T, with a bijection between the leaves of T and $V(G)$.

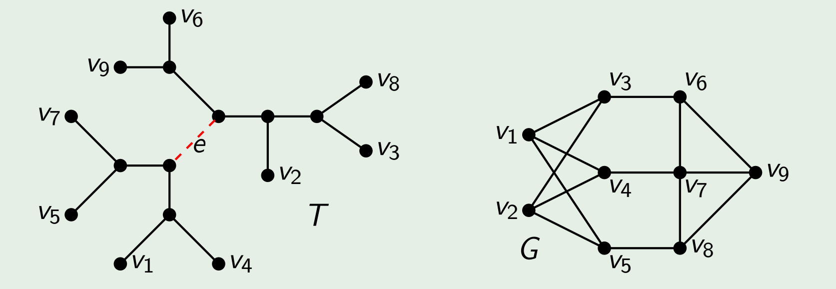


Bipartition $(V_e, \overline{V_e})$ of $V(G)$ displayed by each edge e of T

Define a symmetric width function $f : 2^{V(G)} \to \mathbb{Z}$

Branch decompositions and f-width

A branch decomposition (T, δ) of G is a subcubic tree T, with a bijection between the leaves of T and $V(G)$.



Bipartition $(V_e, \overline{V_e})$ of $V(G)$ displayed by each edge e of T

Define a symmetric width function $f : 2^{V(G)} \rightarrow \mathbb{Z}$

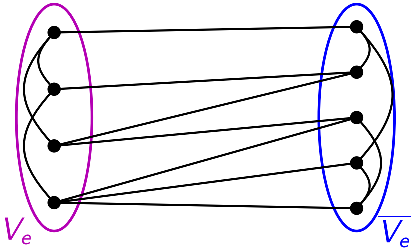
f-width of (T, δ) is max width of a set displayed by an edge of T

f-width of G is the minimum over all branch decompositions

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$



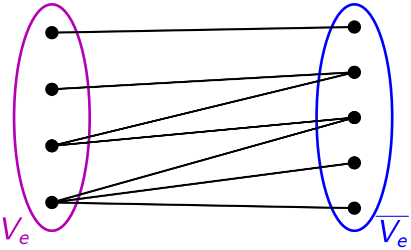
G

f measures the “complexity” of the cut.

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$



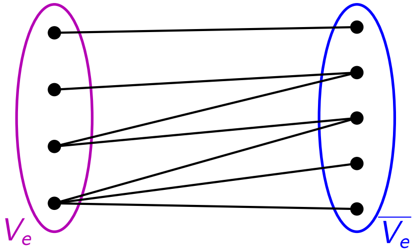
$G[V_e, \overline{V_e}]$ bipartite

f measures the “complexity” of the cut.

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$

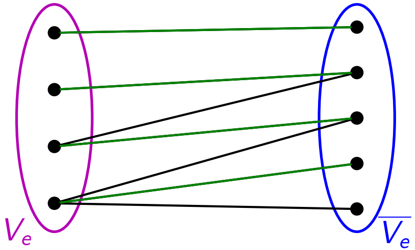


mm-width: f is the maximum size of a matching in $G[V_e, \overline{V_e}]$

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$



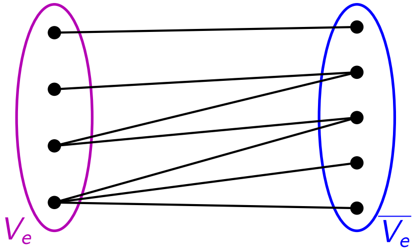
mm-width: f is the maximum size of a matching in $G[V_e, \overline{V_e}]$

$f(V_e) = 4$.

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$

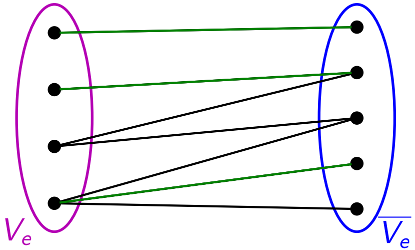


mim-width: f is the max size of an induced matching in $G[V_e, \overline{V_e}]$

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$



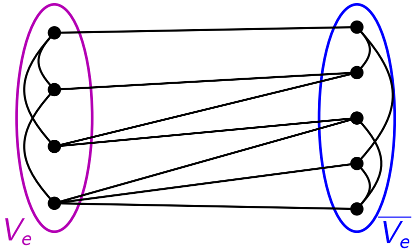
mim-width: f is the max size of an induced matching in $G[V_e, \overline{V_e}]$

$f(V_e) = 3$.

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$

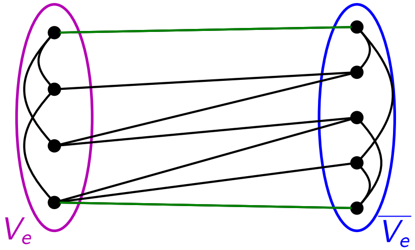


sim-width: f is the maximum size of an induced matching in G (consisting of edges from $G[V_e, \overline{V_e}]$)

Width functions of branch decompositions

Consider a branch decomposition (T, δ) on $V(G)$.

Each edge e of T corresponds to a cut $(V_e, \overline{V_e})$



sim-width: f is the maximum size of an induced matching in G (consisting of edges from $G[V_e, \overline{V_e}]$)

$f(V_e) = 2$.

Comparing more width parameters

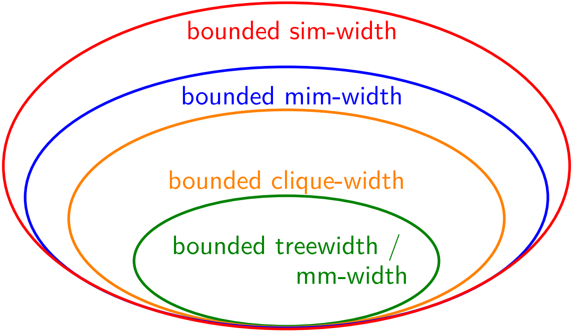
mm-width is equivalent to treewidth:
$\mathrm{mmw}(G) \leq \mathrm{tw}(G) + 1 \leq 3\mathrm{mmw}(G)$

[Vatshelle 2012; Jeong et al. 2018]

mim-width is less restrictive than clique-width
$\mathrm{mimw}(G) \leq \mathrm{cw}(G)$

sim-width is less restrictive than mim-width
$\mathrm{simw}(G) \leq \mathrm{mimw}(G)$ [Kang, Kwon, Strømme, Telle, 2017]

Comparing width parameters



Why mim-width?

interval graphs, permutation graphs have mim-width 1

There is an XP algorithm (runs in $f(w)n^{g(w)}$ time) for a wide range of "locally checkable" problems (IS, k-COLOURING, ...)

[Bui-Xuan, Telle, Vatshelle, 2013]

Generalised to other problems (FVS, LIST k-COLOURING, ...)

Theorem (Bergougnoux, Drier, and Jaffke, 2023)

Any problem expressible in DN *logic of graphs is* XP *parameterised by mim-width, given a branch decomposition*

(there is an $n^{f(w)}$-time algorithm).

Why mim-width?

interval graphs, permutation graphs have mim-width 1

There is an XP algorithm (runs in $f(w)n^{g(w)}$ time) for a wide range of "locally checkable" problems (IS, k-COLOURING, …)

[Bui-Xuan, Telle, Vatshelle, 2013]

Generalised to other problems (FVS, LIST k-COLOURING, …)

Theorem (Bergougnoux, Drier, and Jaffke, 2023)

Any problem expressible in DN *logic of graphs is* XP *parameterised by mim-width, given a branch decomposition*

(there is an $n^{f(w)}$-time algorithm).

Existential MSO_1: quantifiers over sets must be existential and outside any other part of formula

Distance neighbourhood logic (DN): extends existential MSO_1 with predicates for querying about neighbourhoods of sets

Why mim-width?

interval graphs, permutation graphs have mim-width 1

There is an XP algorithm (runs in $f(w)n^{g(w)}$ time) for a wide range of "locally checkable" problems (IS, k-COLOURING, ...)

[Bui-Xuan, Telle, Vatshelle, 2013]

Generalised to other problems (FVS, LIST k-COLOURING, ...)

Theorem (Bergougnoux, Drier, and Jaffke, 2023)

Any problem expressible in DN *logic of graphs is* XP *parameterised by mim-width, given a branch decomposition*

(there is an $n^{f(w)}$-time algorithm).

semitotal dominating set: a dominating set $S \subseteq V(G)$ such that for each $v \in S$ there is distinct $u \in S$ such that $d(u, v) \leq 2$.

$\exists X : |X| \leq m \wedge X \cup N_1^1(X) = \emptyset \wedge X \subseteq N_1^2(X)$

Why mim-width?

interval graphs, permutation graphs have mim-width 1

There is an XP algorithm (runs in $f(w)n^{g(w)}$ time) for a wide range of "locally checkable" problems (IS, k-COLOURING, ...)

[Bui-Xuan, Telle, Vatshelle, 2013]

Generalised to other problems (FVS, LIST k-COLOURING, ...)

Theorem (Bergougnoux, Drier, and Jaffke, 2023)

Any problem expressible in DN *logic of graphs is* XP *parameterised by mim-width, given a branch decomposition*

(there is an $n^{f(w)}$-time algorithm).

Obtaining decompositions

treewidth/clique-width meta theorems don't require decomposition as input

Theorem (Bodlaender 2006)

For fixed k, there is a linear-time algorithm that finds a tree decomposition of width $\leq k$ or determines $\mathrm{tw}(G) > k$.

Theorem (Oum and Seymour 2007)

For fixed k, if f is submodular, there is an $O(n^{g(k)} \log n)$-time algorithm that finds a branch decomposition of f-width $3k+1$ or determines f-width is more than k.

Obtaining decompositions

treewidth/clique-width meta theorems don't require decomposition as input

Theorem (Bodlaender 2006)

For fixed k, there is a linear-time algorithm that finds a tree decomposition of width $\leq k$ or determines $\mathrm{tw}(G) > k$.

Theorem (Oum and Seymour 2007)

For fixed k, if f is submodular, there is an $O(n^{g(k)} \log n)$-time algorithm that finds a branch decomposition of f-width $3k+1$ or determines f-width is more than k.

mim-width function is not submodular (and NP-hard to compute)

Open problem: is there an XP (approximation) algorithm for computing a branch decomposition of mim-width at most k?

Why sim-width?

Bounded for chordal graphs.

Of theoretical interest, but few algorithmic applications.

List k-colouring is polynomial-time solvable for graphs with bounded sim-width, given a decomposition

[Munaro and Yang 2023]

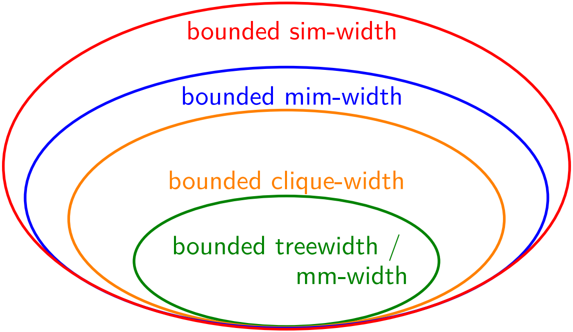
$\mathrm{simw}(G) \leq \mathrm{mimw}(G)$

$\mathrm{simw}(G/e) \leq \mathrm{simw}(G)$.

For K_t-free graphs, there exists f s.t. $\mathrm{mimw}(G) \leq f(\mathrm{simw}(G))$.

[Kang, Kwon, Strømme, Telle, 2017]

Comparing width parameters, revisited



Comparing width parameters, revisited

sim-width

$\downarrow$

mim-width

$\downarrow$

clique-width

$\downarrow$

treewidth

Comparing width parameters, revisited

sim-width

↓

mim-width

↓

clique-width

↓

treewidth

Theorem (B., Munaro, Paulusma, Yang, 2023+)

For any t, treewidth, clique-width, mim-width, and sim-width are equivalent when restricted to $K_{t,t}$-subgraph free graphs.

Comparing width parameters, revisited

For $K_{t,t}$-subgraph-free graphs:

sim-width

$\updownarrow$

mim-width

$\updownarrow$

clique-width

$\updownarrow$

treewidth

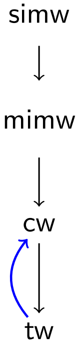
Theorem (B., Munaro, Paulusma, Yang, 2023+)

For any t, treewidth, clique-width, mim-width, and sim-width are equivalent when restricted to $K_{t,t}$-subgraph free graphs.

Proof of equivalence for $K_{t,t}$-subgraph-free graphs

For $K_{t,t}$-subgraph free: $\mathrm{tw}(G) \leq 3(n-1) \cdot \mathrm{cw}(G) - 1$.

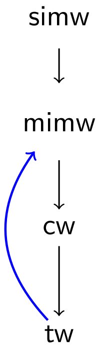
[Gurski and Wanke 2000]



Proof of equivalence for $K_{t,t}$-subgraph-free graphs

For $K_{t,t}$-subgraph free: $\mathrm{tw}(G) \leq 3(n-1) \cdot \mathrm{cw}(G) - 1$.

[Gurski and Wanke 2000]



Lemma

Given t and p, there exists $N(t,p)$ such that every bipartite graph with a matching of size $N(t,p)$ and having no $K_{t,t}$-subgraph contains an induced matching of size p.

For $K_{t,t}$-subgraph-free: $\mathrm{mmw}(G) < N(t, \mathrm{mimw}(G))$.

Proof of equivalence for $K_{t,t}$-subgraph-free graphs

For $K_{t,t}$-subgraph free: $\mathrm{tw}(G) \leq 3(n-1) \cdot \mathrm{cw}(G) - 1$.

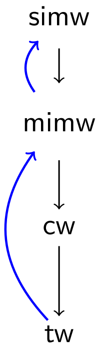
[Gurski and Wanke 2000]

Lemma

Given t and p, there exists $N(t,p)$ such that every bipartite graph with a matching of size $N(t,p)$ and having no $K_{t,t}$-subgraph contains an induced matching of size p.

For $K_{t,t}$-subgraph-free: $\mathrm{mmw}(G) < N(t, \mathrm{mimw}(G))$.

Another Ramsey-theoretic argument for $K_{t,t}$-subgraph-free graphs gives $\mathrm{mimw}(G) \leq f(\mathrm{simw}(G))$.



Line graphs

Theorem (B., Munaro, Paulusma, Yang, 2023+)

Let $\mathcal{G}$ be a class of graphs and let $L(\mathcal{G})$ be the class of line graphs of graphs in $\mathcal{G}$. The following are equivalent:

1. *The class $\mathcal{G}$ has bounded treewidth.*
2. *The class $L(\mathcal{G})$ has bounded clique-width.*
3. *The class $L(\mathcal{G})$ has bounded mim-width.*
4. *The class $L(\mathcal{G})$ has bounded sim-width.*

Line graphs

Theorem (B., Munaro, Paulusma, Yang, 2023+)

Let $\mathcal{G}$ be a class of graphs and let $L(\mathcal{G})$ be the class of line graphs of graphs in $\mathcal{G}$. The following are equivalent:

1. *The class $\mathcal{G}$ has bounded treewidth.*
2. *The class $L(\mathcal{G})$ has bounded clique-width.*
3. *The class $L(\mathcal{G})$ has bounded mim-width.*
4. *The class $L(\mathcal{G})$ has bounded sim-width.*

$$\frac{\mathrm{cw}(L(G)) - 3}{2} < \mathrm{tw}(G) < 4\mathrm{cw}(L(G))$$

[Gurski and Wanke 2007]

$$\mathrm{simw}(G) \leq \mathrm{mimw}(G) \leq \mathrm{cw}(G)$$

So suffices to prove $\mathrm{tw}(G) \leq f(\mathrm{simw}(L(G)))$ for some function f.

Proof of equivalence for line graphs

$\mathrm{simw}(G - v) \leq \mathrm{simw}(G)$
$\mathrm{simw}(G/e) \leq \mathrm{simw}(G)$ [Kang et al. 2017]

Lemma (B., Munaro, Paulusma, Yang, 2023+)

$\mathrm{simw}(L(G/e)) \leq \mathrm{simw}(L(G))$

Proof of equivalence for line graphs

$\mathrm{simw}(G - v) \leq \mathrm{simw}(G)$
$\mathrm{simw}(G/e) \leq \mathrm{simw}(G)$ [Kang et al. 2017]

Lemma (B., Munaro, Paulusma, Yang, 2023+)

$\mathrm{simw}(L(G/e)) \leq \mathrm{simw}(L(G))$

Lemma

$\mathrm{tw}(G) \leq f(\mathrm{simw}(L(G)))$ *for some function f.*

Proof.

if $\mathrm{tw}(G)$ large, then G has a large grid minor H

$\mathrm{simw}(L(G)) \geq \mathrm{simw}(L(H))$

since $L(H)$ is $K_{6,6}$-subgraph-free, there are g and h s.t.
$g(\mathrm{simw}(L(H))) \geq \mathrm{tw}(L(H))$ and $h(\mathrm{tw}(L(H))) \geq \mathrm{tw}(G)$. □

Tree-independence number

What if cliques are the only obstruction to having small treewidth?

Independence number of a tree decomposition: the maximum size of an independent set induced by a bag

Tree-independence number of a graph G, denoted $\text{tree-}\alpha(G)$: the minimum independence number over all tree decompositions.

$\text{simw}(G) \leq \text{tree-}\alpha(G)$ [Munaro and Yang 2023]

$\text{tree-}\alpha(G) \leq \text{tw}(G) + 1$ [Dallard, Milanič, Štorgel, 2024]

Why tree-independence number?

Various algorithmic results for packing problems (IS, FVS, $\ldots$)

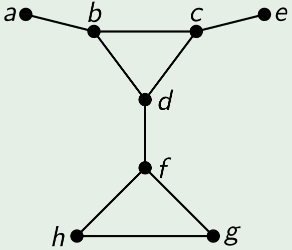
Theorem (Dallard, Fomin, Golovach, Korhonen, Milanič 2022+)

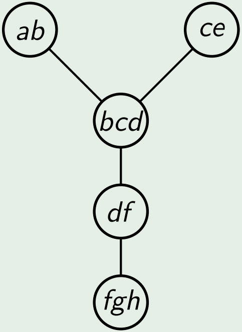
There is an approximation algorithm for computing a tree decomposition of bounded independence number.

Conjecture (Dallard, Milanič, Štorgel 2022+)

A hereditary class has bounded tree-independence number iff there exists a function f such that $\mathrm{tw}(G) \leq f(\omega(G))$.

Tree-independence number: an example

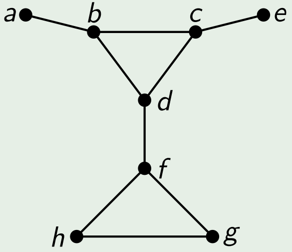


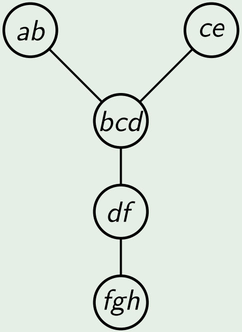


This decomposition has independence number 1.

$$\text{tree-}\alpha(G) \leq 1$$

Tree-independence number: an example





This decomposition has independence number 1.

$$\text{tree-}\alpha(G) = 1$$

Comparing the class of all graphs

sim-width

↓

mim-width

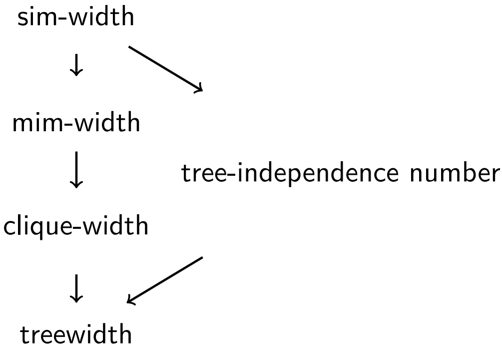
↓

clique-width

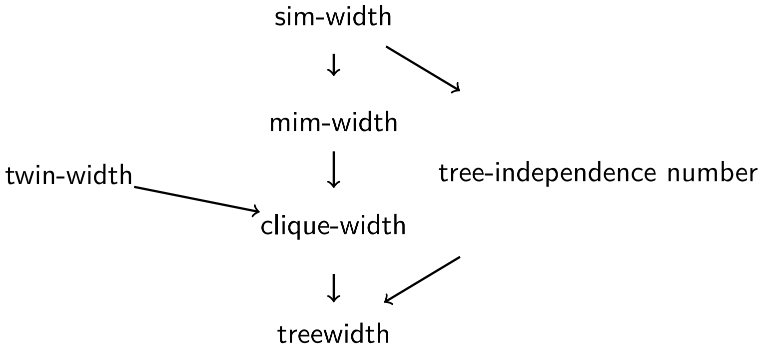
↓

treewidth

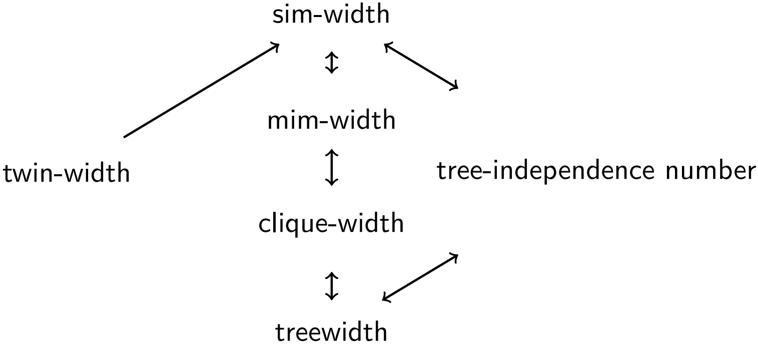
Comparing the class of all graphs



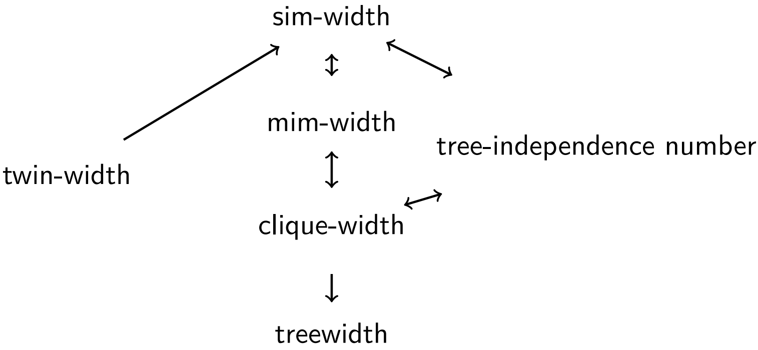
Comparing the class of all graphs



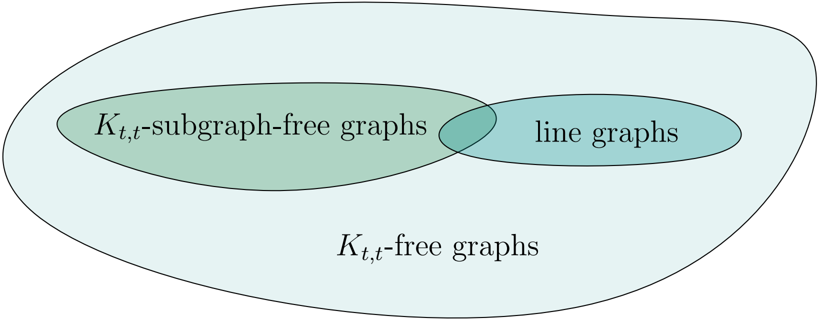
Comparing $K_{t,t}$-subgraph-free graphs



Comparing line graphs

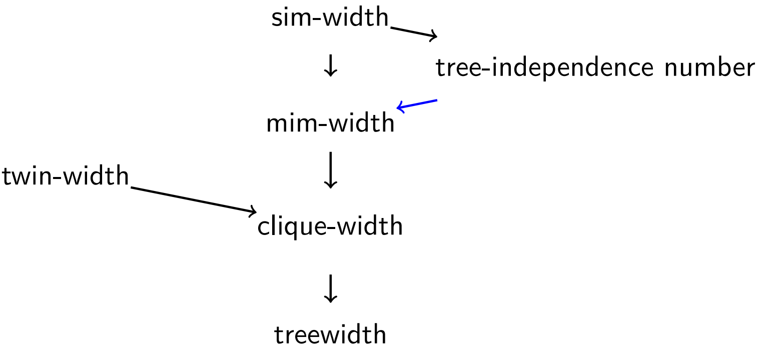


Comparing $K_{t,t}$-free graphs



$K_{t,t}$-free: no induced subgraph isomorphic to $K_{t,t}$.

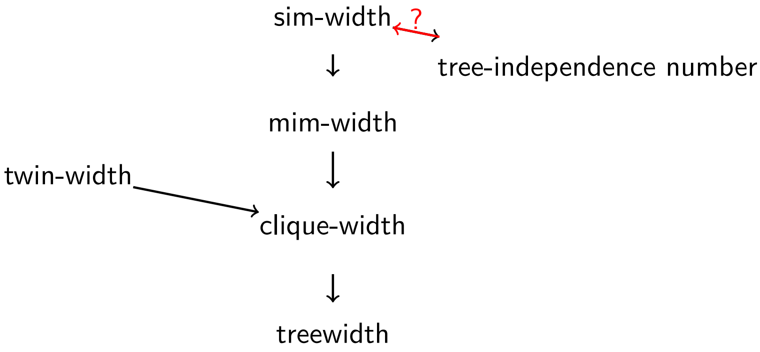
Comparing $K_{t,t}$-free graphs



Theorem (B., Munaro, Paulusma, Yang, 2023+)

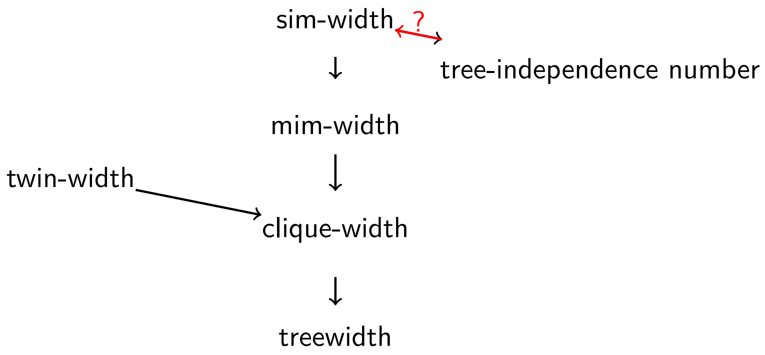
Given a $K_{s,t}$-free graph G and a decomposition of mim-width w, we can construct a tree decomposition of G with independence number at most $6(2^{t+w-1} + sw^{t+1})$ in $O(n^{sw^t+4})$-time.

Comparing $K_{t,t}$-free graphs



Open problem: is tree-independence number less restrictive than sim-width for $K_{t,t}$-free graphs?

Comparing $K_{t,t}$-free graphs



Open problem: is tree-independence number less restrictive than sim-width for $K_{t,t}$-free graphs?

Yes. [Abrishami, Briański, Czyżewska, McCarty, Milanič, and Rzążewski]

Open problems

Is there an XP algorithm, parameterized by k, that either decides that $\mathrm{mimw}(G) > k$ (or $\mathrm{simw}(G) > k$), or outputs a decomposition of G of mim-width (or sim-width) at most $f(k)$?

Open problems

Is there an XP algorithm, parameterized by k, that either decides that $\mathrm{mimw}(G) > k$ (or $\mathrm{simw}(G) > k$), or outputs a decomposition of G of mim-width (or sim-width) at most $f(k)$?

$$\frac{\mathrm{tw}(G)+1}{4} \leq \mathrm{cw}(L(G)) \leq 2\mathrm{tw}(G)+2$$

[Gurski and Wanke 2007]

$$\left\lfloor \frac{\mathrm{bw}(G)}{25} \right\rfloor \leq \mathrm{mimw}(L(G)) \leq \mathrm{bw}(G)$$

[B., Munaro, Paulusma, Yang, 2023+]

Open: similar bounds for sim-width? tree-independence number?

Open problems

If G is d-degenerate with a matching of size μ, then G has an induced matching of size at least $\mu/(4d-1)$.

Can we do better? (Can't do better than $\mu/2d$.)

Open problems

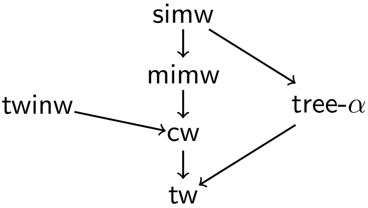
If G is d-degenerate with a matching of size μ, then G has an induced matching of size at least $\mu/(4d-1)$.

Can we do better? (Can't do better than $\mu/2d$.)

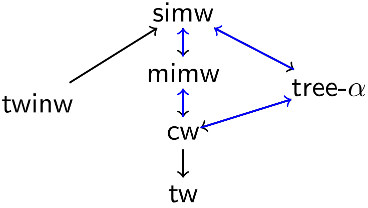
Find an asymptotically optimal upper bound on tree-independence number in terms of clique-width and the largest induced $K_{t,t}$.

Summary

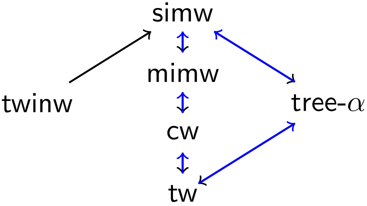
All graphs:



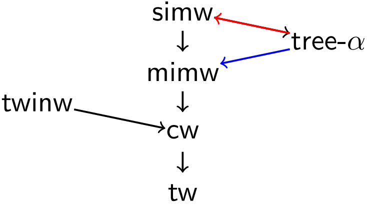
Line graphs:



$K_{t,t}$-subgraph free:

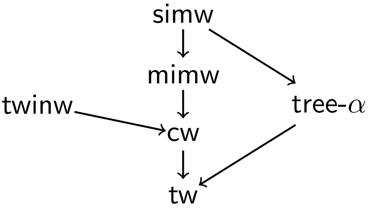


$K_{t,t}$-free:

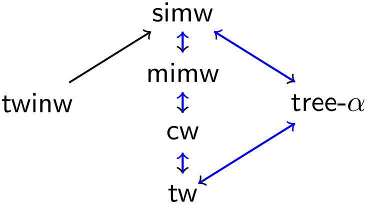


Summary

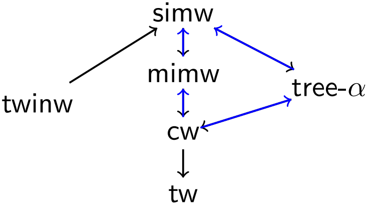
All graphs:



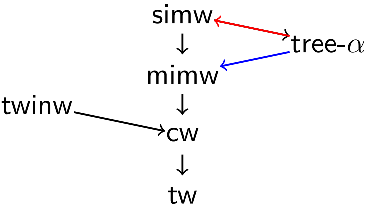
$K_{t,t}$-subgraph free:



Line graphs:



$K_{t,t}$-free:



Thanks for your attention.