Finding sparse induced subgraphs on graphs of bounded induced matching treewidth

Hans L. Bodlaender[1], Fedor V. Fomin[2], and Tuukka Korhonen

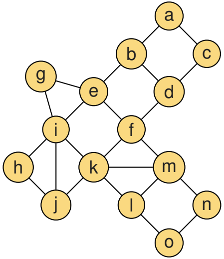
UNIVERSITY OF COPENHAGEN

[1]Utrecht University, [2]University of Bergen

SODA 2026

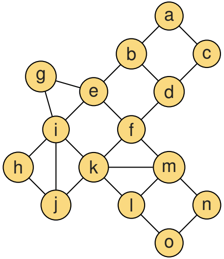
12 January 2026

Treewidth

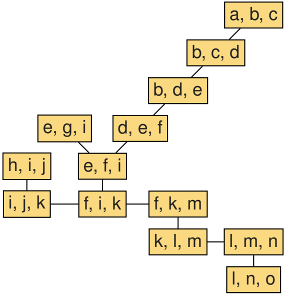


Graph G

Treewidth

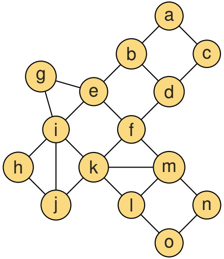


Graph G

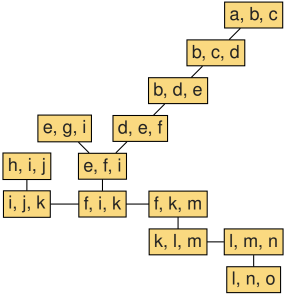


A tree decomposition of G

Treewidth



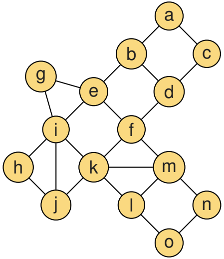
Graph G



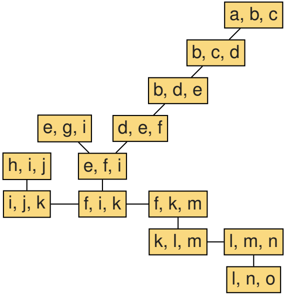
A tree decomposition of G

1. Every vertex should be in a bag

Treewidth



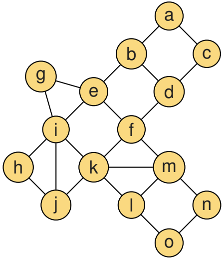
Graph G



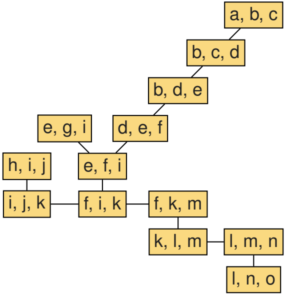
A tree decomposition of G

1. Every vertex should be in a bag
2. Every edge should be in a bag

Treewidth



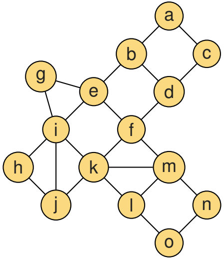
Graph G



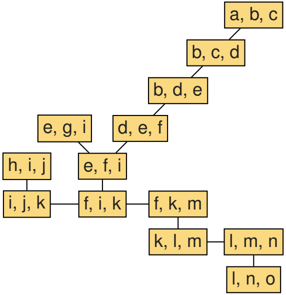
A tree decomposition of G

1. Every vertex should be in a bag
2. Every edge should be in a bag
3. For every vertex v, the bags containing v should form a connected subtree

Treewidth



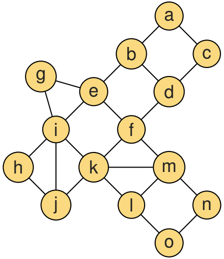
Graph G



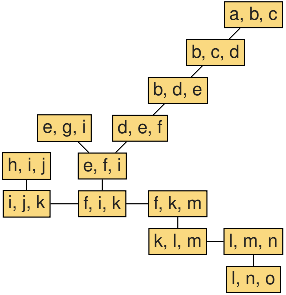
A tree decomposition of G

1. Every vertex should be in a bag
2. Every edge should be in a bag
3. For every vertex v, the bags containing v should form a connected subtree
4. Width = maximum bag size -1

Treewidth



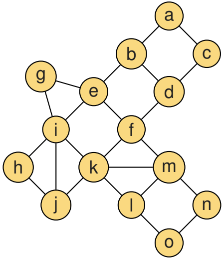
Graph G



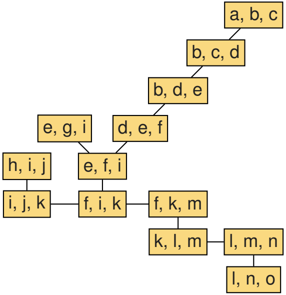
A tree decomposition of G
Width = 2

1. Every vertex should be in a bag
2. Every edge should be in a bag
3. For every vertex v, the bags containing v should form a connected subtree
4. Width = maximum bag size -1

Treewidth



Graph G

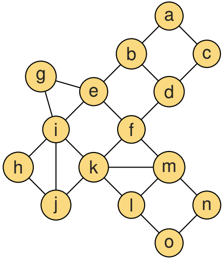


A tree decomposition of G

Width = 2

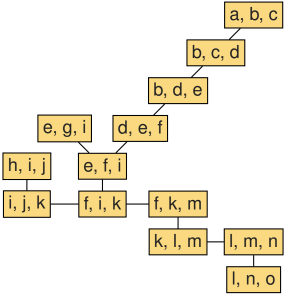
1. Every vertex should be in a bag
2. Every edge should be in a bag
3. For every vertex v, the bags containing v should form a connected subtree
4. Width = maximum bag size -1
5. Treewidth of G = minimum width of tree decomposition of G

Treewidth



Graph G

Treewidth 2

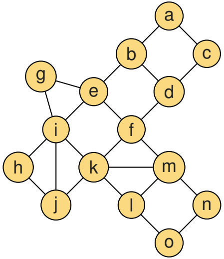


A tree decomposition of G

Width = 2

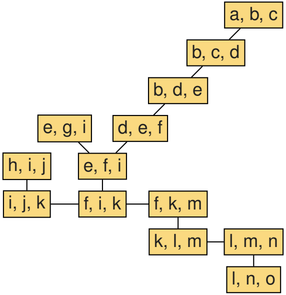
1. Every vertex should be in a bag
2. Every edge should be in a bag
3. For every vertex v, the bags containing v should form a connected subtree
4. Width = maximum bag size -1
5. Treewidth of G = minimum width of tree decomposition of G

Treewidth



Graph G

Treewidth 2



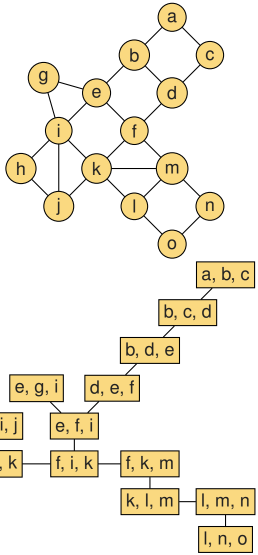
A tree decomposition of G

Width = 2

1. Every vertex should be in a bag
2. Every edge should be in a bag
3. For every vertex v, the bags containing v should form a connected subtree
4. Width = maximum bag size -1
5. Treewidth of G = minimum width of tree decomposition of G

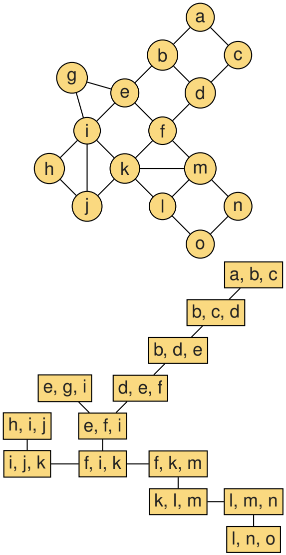
[Robertson & Seymour'84, Arnborg & Proskurowski'89, Bertele & Brioschi'72, Halin'76]

Why treewidth?



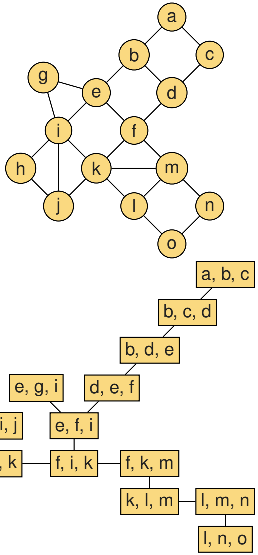
Why treewidth?

- When given a graph with a tree decomposition of bounded width, many NP-hard graph problems can be solved in polynomial time



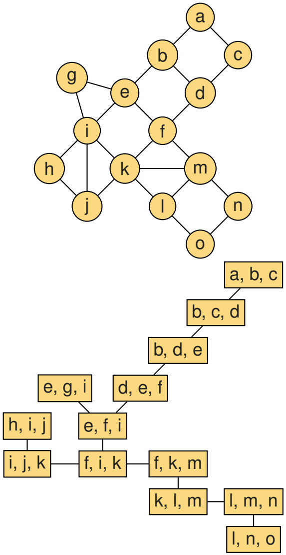
Why treewidth?

- When given a graph with a tree decomposition of bounded width, many NP-hard graph problems can be solved in polynomial time
- Example: Maximum independent set in $\mathcal{O}(2^k \cdot n)$ time on treewidth-k graphs by dynamic programming



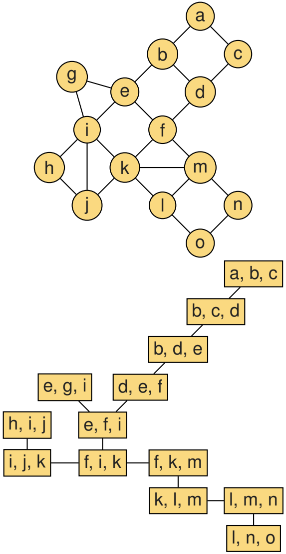
Why treewidth?

- When given a graph with a tree decomposition of bounded width, many NP-hard graph problems can be solved in polynomial time
- Example: Maximum independent set in $\mathcal{O}(2^k \cdot n)$ time on treewidth-k graphs by dynamic programming
- Courcelle's Theorem gives $f(k) \cdot n$ time algorithms for all **CMSO**$_2$-definable problems



Why treewidth?

- When given a graph with a tree decomposition of bounded width, many NP-hard graph problems can be solved in polynomial time
- Example: Maximum independent set in $\mathcal{O}(2^k \cdot n)$ time on treewidth-k graphs by dynamic programming
- Courcelle's Theorem gives $f(k) \cdot n$ time algorithms for all **CMSO**$_2$-definable problems



Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

- Treewidth: Measure $\max |B_x| - 1$

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

- Treewidth: Measure $\max |B_x| - 1$
- Tree-independence-number: Measure $\max \alpha(G[B_x])$, where α is the maximum size of independent set [Yolov, SODA'18]

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

- Treewidth: Measure $\max |B_x| - 1$
- Tree-independence-number: Measure $\max \alpha(G[B_x])$, where α is the maximum size of independent set [Yolov, SODA'18]
  - Observation: Maximum independent set in time $n^{\mathcal{O}(k)}$

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

- Treewidth: Measure $\max |B_x| - 1$
- Tree-independence-number: Measure $\max \alpha(G[B_x])$, where α is the maximum size of independent set [Yolov, SODA'18]
  - Observation: Maximum independent set in time $n^{\mathcal{O}(k)}$
- Induced matching treewidth: Measure $\max \mu(B_x)$, where $\mu(B_x)$ is the maximum size of an induced matching M in G, so that every edge $uv \in M$ has $|\{u, v\} \cap B_x| \geq 1$. [Yolov, SODA'18]

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

- Treewidth: Measure $\max |B_x| - 1$
- Tree-independence-number: Measure $\max \alpha(G[B_x])$, where α is the maximum size of independent set [Yolov, SODA'18]
  - Observation: Maximum independent set in time $n^{\mathcal{O}(k)}$
- Induced matching treewidth: Measure $\max \mu(B_x)$, where $\mu(B_x)$ is the maximum size of an induced matching M in G, so that every edge $uv \in M$ has $|\{u, v\} \cap B_x| \geq 1$. [Yolov, SODA'18]
  - Non-trivial result: Maximum independent set in time $n^{\mathcal{O}(k)}$ [Yolov, SODA'18]

More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

- Treewidth: Measure $\max |B_x| - 1$
- Tree-independence-number: Measure $\max \alpha(G[B_x])$, where α is the maximum size of independent set [Yolov, SODA'18]
  - Observation: Maximum independent set in time $n^{\mathcal{O}(k)}$
- Induced matching treewidth: Measure $\max \mu(B_x)$, where $\mu(B_x)$ is the maximum size of an induced matching M in G, so that every edge $uv \in M$ has $|\{u, v\} \cap B_x| \geq 1$. [Yolov, SODA'18]
  - Non-trivial result: Maximum independent set in time $n^{\mathcal{O}(k)}$ [Yolov, SODA'18]
- Observation:
  - $\text{tree-}\mu(G) \leq \text{tree-}\alpha(G) \leq \text{tw}(G) + 1$

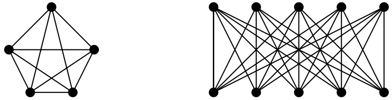
More exotic widths

Question

Can we generalize the definition of width, while retaining useful algorithmic applications?

Let $(T, B_{x \in V(T)})$ be a tree decomposition.

- Treewidth: Measure $\max |B_x| - 1$
- Tree-independence-number: Measure $\max \alpha(G[B_x])$, where α is the maximum size of independent set [Yolov, SODA'18]
  - Observation: Maximum independent set in time $n^{\mathcal{O}(k)}$
- Induced matching treewidth: Measure $\max \mu(B_x)$, where $\mu(B_x)$ is the maximum size of an induced matching M in G, so that every edge $uv \in M$ has $|\{u, v\} \cap B_x| \geq 1$. [Yolov, SODA'18]
  - Non-trivial result: Maximum independent set in time $n^{\mathcal{O}(k)}$ [Yolov, SODA'18]
- Observation:
  - $\text{tree-}\mu(G) \leq \text{tree-}\alpha(G) \leq \text{tw}(G) + 1$



Our result

We generalize the independent set algorithm of [Yolov, SODA'18] to a meta-theorem

Our result

We generalize the independent set algorithm of [Yolov, SODA'18] to a meta-theorem

Theorem (This paper)

Given an n-vertex graph G with tree-$\mu(G) \leq k$, a **CMSO$_2$**-sentence Φ, and an integer w, we can in time $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ find a maximum-size set $X \subseteq V(G)$ so that

1. $\mathrm{tw}(G[X]) \leq w$
2. $G[X] \models \Phi$

where f is a computable function.

Our result

We generalize the independent set algorithm of [Yolov, SODA'18] to a meta-theorem

Theorem (This paper)

Given an n-vertex graph G with tree-$\mu(G) \leq k$, a $\mathbf{CMSO}_2$-sentence Φ, and an integer w, we can in time $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ find a maximum-size set $X \subseteq V(G)$ so that

1. $\text{tw}(G[X]) \leq w$
2. $G[X] \models \Phi$

where f is a computable function.

Implies $f(k) \cdot n^{\mathcal{O}(k)}$ time algorithms for

- maximum independent set (given by [Yolov'18])
- minimum feedback vertex set (given by [Lima, Milanič, Muršič, Okrasa, Rzążewski, and Štorgel, '24])

Our result

We generalize the independent set algorithm of [Yolov, SODA'18] to a meta-theorem

Theorem (This paper)

Given an n-vertex graph G with tree-$\mu(G) \leq k$, a $\mathbf{CMSO}_2$-sentence Φ, and an integer w, we can in time $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ find a maximum-size set $X \subseteq V(G)$ so that

1. $\mathrm{tw}(G[X]) \leq w$
2. $G[X] \models \Phi$

where f is a computable function.

Implies $f(k) \cdot n^{\mathcal{O}(k)}$ time algorithms for

- maximum independent set (given by [Yolov'18])
- minimum feedback vertex set (given by [Lima, Milanič, Muršič, Okrasa, Rzążewski, and Štorgel, '24])
- maximum induced matching, longest induced path, longest induced cycle, minimum planar vertex deletion, maximum cubic induced subgraph, etc... [New]

Our result

We generalize the independent set algorithm of [Yolov, SODA'18] to a meta-theorem

Theorem (This paper)

Given an n-vertex graph G with tree-$\mu(G) \leq k$, a $\mathbf{CMSO_2}$-sentence Φ, and an integer w, we can in time $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ find a maximum-size set $X \subseteq V(G)$ so that

1. $\text{tw}(G[X]) \leq w$
2. $G[X] \models \Phi$

where f is a computable function.

Implies $f(k) \cdot n^{\mathcal{O}(k)}$ time algorithms for

- maximum independent set (given by [Yolov'18])
- minimum feedback vertex set (given by [Lima, Milanič, Muršič, Okrasa, Rzążewski, and Štorgel, '24])
- maximum induced matching, longest induced path, longest induced cycle, minimum planar vertex deletion, maximum cubic induced subgraph, etc... [New]

Answers an open problem of [Lima, Milanič, Muršič, Okrasa, Rzążewski, and Štorgel, '24], who gave a similar theorem for tree-independence number

The algorithm

Step 1: Containers

Yolov's idea:

Lemma (Yolov'18)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $n^{\mathcal{O}(k)}$ possible intersections $B \cap I$, where I is a maximal independent set of G.

Step 1: Containers

Yolov's idea:

Lemma (Yolov'18)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $n^{\mathcal{O}(k)}$ possible intersections $B \cap I$, where I is a maximal independent set of G.

$\Rightarrow$ leads immediately to an $n^{\mathcal{O}(k)}$ time algorithm for maximum independent set

Step 1: Containers

Yolov's idea:

Lemma (Yolov'18)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $n^{\mathcal{O}(k)}$ possible intersections $B \cap I$, where I is a maximal independent set of G.

$\Rightarrow$ leads immediately to an $n^{\mathcal{O}(k)}$ time algorithm for maximum independent set

Lemma (This work)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $f(k, w, \Phi) \cdot n^{\mathcal{O}(kw)}$ possible intersections $B \cap X$, where X is a maximal set $X \subseteq V(G)$ so that $\mathrm{tw}(G[X]) \leq w$ and $G[X] \models \Phi$.

Step 1: Containers

Yolov's idea:

Lemma (Yolov'18)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $n^{\mathcal{O}(k)}$ possible intersections $B \cap I$, where I is a maximal independent set of G.

$\Rightarrow$ leads immediately to an $n^{\mathcal{O}(k)}$ time algorithm for maximum independent set

Lemma (This work)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $f(k, w, \Phi) \cdot n^{\mathcal{O}(kw)}$ possible intersections $B \cap X$, where X is a maximal set $X \subseteq V(G)$ so that $\mathrm{tw}(G[X]) \leq w$ and $G[X] \models \Phi$.

Main ideas of the proof:

Step 1: Containers

Yolov's idea:

Lemma (Yolov'18)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $n^{\mathcal{O}(k)}$ possible intersections $B \cap I$, where I is a maximal independent set of G.

$\Rightarrow$ leads immediately to an $n^{\mathcal{O}(k)}$ time algorithm for maximum independent set

Lemma (This work)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $f(k, w, \Phi) \cdot n^{\mathcal{O}(kw)}$ possible intersections $B \cap X$, where X is a maximal set $X \subseteq V(G)$ so that $\mathrm{tw}(G[X]) \leq w$ and $G[X] \models \Phi$.

Main ideas of the proof:

- edges of $G[X]$ intersecting B must have vertex cover of size $\mathcal{O}(kw)$

Step 1: Containers

Yolov's idea:

Lemma (Yolov'18)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $n^{\mathcal{O}(k)}$ possible intersections $B \cap I$, where I is a maximal independent set of G.

$\Rightarrow$ leads immediately to an $n^{\mathcal{O}(k)}$ time algorithm for maximum independent set

Lemma (This work)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $f(k, w, \Phi) \cdot n^{\mathcal{O}(kw)}$ possible intersections $B \cap X$, where X is a maximal set $X \subseteq V(G)$ so that $\mathrm{tw}(G[X]) \leq w$ and $G[X] \models \Phi$.

Main ideas of the proof:

- edges of $G[X]$ intersecting B must have vertex cover of size $\mathcal{O}(kw)$
- guess the vertex cover

Step 1: Containers

Yolov's idea:

Lemma (Yolov'18)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $n^{\mathcal{O}(k)}$ possible intersections $B \cap I$, where I is a maximal independent set of G.

$\Rightarrow$ leads immediately to an $n^{\mathcal{O}(k)}$ time algorithm for maximum independent set

Lemma (This work)

For a set $B \subseteq V(G)$ with $\mu(B) \leq k$, there are at most $f(k, w, \Phi) \cdot n^{\mathcal{O}(kw)}$ possible intersections $B \cap X$, where X is a maximal set $X \subseteq V(G)$ so that $\mathrm{tw}(G[X]) \leq w$ and $G[X] \models \Phi$.

Main ideas of the proof:

- edges of $G[X]$ intersecting B must have vertex cover of size $\mathcal{O}(kw)$
- guess the vertex cover
- use Yolov's lemma, a "kicking out"-lemma, and properties of **CMSO$_2$** to guess the rest

Step 2: Inner tree decomposition

How to figure our whether $G[X] \models \Phi$?

Step 2: Inner tree decomposition

How to figure our whether $G[X] \models \Phi$?

Lemma (Informal)

Let T be a nice tree decomposition of G with $\mu(T) \leq k$, and $X \subseteq V(G)$ a set with $\text{tw}(G[X]) \leq w$. There is a tree decomposition T_X of $G[X]$ of width $\mathcal{O}(kw^2)$, whose structure "follows" T.

Step 2: Inner tree decomposition

How to figure our whether $G[X] \models \Phi$?

Lemma (Informal)

Let T be a nice tree decomposition of G with $\mu(T) \leq k$, and $X \subseteq V(G)$ a set with $\text{tw}(G[X]) \leq w$. There is a tree decomposition T_X of $G[X]$ of width $\mathcal{O}(kw^2)$, whose structure "follows" T.

- The trees of T and T_X are the same

Step 2: Inner tree decomposition

How to figure our whether $G[X] \models \Phi$?

Lemma (Informal)

Let T be a nice tree decomposition of G with $\mu(T) \leq k$, and $X \subseteq V(G)$ a set with $\text{tw}(G[X]) \leq w$. There is a tree decomposition T_X of $G[X]$ of width $\mathcal{O}(kw^2)$, whose structure "follows" T.

- The trees of T and T_X are the same
- There is a partition (X_1, X_2) of X so that:

Step 2: Inner tree decomposition

How to figure our whether $G[X] \models \Phi$?

Lemma (Informal)

Let T be a nice tree decomposition of G with $\mu(T) \leq k$, and $X \subseteq V(G)$ a set with $\text{tw}(G[X]) \leq w$. There is a tree decomposition T_X of $G[X]$ of width $\mathcal{O}(kw^2)$, whose structure “follows” T.

- The trees of T and T_X are the same
- There is a partition (X_1, X_2) of X so that:
  - Vertices in X_1 extend their appearances upwards compared to T

Step 2: Inner tree decomposition

How to figure our whether $G[X] \models \Phi$?

Lemma (Informal)

Let T be a nice tree decomposition of G with $\mu(T) \leq k$, and $X \subseteq V(G)$ a set with $\mathrm{tw}(G[X]) \leq w$. There is a tree decomposition T_X of $G[X]$ of width $\mathcal{O}(kw^2)$, whose structure “follows” T.

- The trees of T and T_X are the same
- There is a partition (X_1, X_2) of X so that:
  - Vertices in X_1 extend their appearances upwards compared to T
  - Vertices in X_2 form an independent set and each appears only in its “top bag”

Step 2: Inner tree decomposition

How to figure our whether $G[X] \models \Phi$?

Lemma (Informal)

Let T be a nice tree decomposition of G with $\mu(T) \leq k$, and $X \subseteq V(G)$ a set with $\mathrm{tw}(G[X]) \leq w$. There is a tree decomposition T_X of $G[X]$ of width $\mathcal{O}(kw^2)$, whose structure "follows" T.

- The trees of T and T_X are the same
- There is a partition (X_1, X_2) of X so that:
  - Vertices in X_1 extend their appearances upwards compared to T
  - Vertices in X_2 form an independent set and each appears only in its "top bag"
- This allows to run a dynamic programming on T, that simultaneously:
  - Guesses the set X
  - Guesses the decomposition T_X
  - Evaluates Φ on $G[X]$ by using T_X

Conclusion

- An $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ time algorithm for finding maximum induced subgraph of treewidth $\leq w$ satisfying Φ on graphs of tree-$\mu(G) \leq k$

Conclusion

- An $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ time algorithm for finding maximum induced subgraph of treewidth $\leq w$ satisfying Φ on graphs of tree-$\mu(G) \leq k$
- Open problems:

Conclusion

- An $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ time algorithm for finding maximum induced subgraph of treewidth $\leq w$ satisfying Φ on graphs of tree-$\mu(G) \leq k$
- Open problems:
  - More general meta-theorem to also capture c-coloring for constant c?

Conclusion

- An $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ time algorithm for finding maximum induced subgraph of treewidth $\leq w$ satisfying Φ on graphs of tree-$\mu(G) \leq k$
- Open problems:
  - More general meta-theorem to also capture c-coloring for constant c?
  - Even more general parameters for tree decompositions?

Conclusion

- An $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ time algorithm for finding maximum induced subgraph of treewidth $\leq w$ satisfying Φ on graphs of tree-$\mu(G) \leq k$
- Open problems:
  - More general meta-theorem to also capture c-coloring for constant c?
  - Even more general parameters for tree decompositions?
    - For example: Measuring a largest induced collection of P_3:s intersecting a bag?

Conclusion

- An $f(k, w, |\Phi|) \cdot n^{\mathcal{O}(kw^2)}$ time algorithm for finding maximum induced subgraph of treewidth $\leq w$ satisfying Φ on graphs of tree-$\mu(G) \leq k$
- Open problems:
  - More general meta-theorem to also capture c-coloring for constant c?
  - Even more general parameters for tree decompositions?
    - For example: Measuring a largest induced collection of P_3:s intersecting a bag?

Thank you!