Computing cutwidth and pathwidth of semi-complete digraphs via degree orderings

Michał Pilipczuk[1]

[1] Department of Informatics, University of Bergen, Norway

STACS 2013, February 28^{th}

Motivation

Containment testing

Input: (di)graphs G and H

Question: Is H contained in G as a minor/topological minor/immersion/...?

Algorithms

- **XP**: polynomial for every fixed H, for instance $O(|G|^{O(|H|)})$.

Algorithms

- **XP**: polynomial for every fixed H, for instance $O(|G|^{O(|H|)})$.
- **FPT**: running time of form $f(|H|) \cdot |G|^c$ for **constant** c.

Algorithms

- **XP**: polynomial for every fixed H, for instance $O(|G|^{O(|H|)})$.
- **FPT**: running time of form $f(|H|) \cdot |G|^c$ for **constant** c.
- For undirected G and H and all these relations, there are $f(|H|) \cdot |G|^3$ algorithms.

Algorithms

- **XP**: polynomial for every fixed H, for instance $O(|G|^{O(|H|)})$.
- **FPT**: running time of form $f(|H|) \cdot |G|^c$ for **constant** c.
- For undirected G and H and all these relations, there are $f(|H|) \cdot |G|^3$ algorithms.
- In directed setting, the problems are already hard for fixed graphs H of constant size.

Tournaments

- It turns out that many problems can be solved efficiently in the class of **tournaments**: work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.

Tournaments

- It turns out that many problems can be solved efficiently in the class of **tournaments**: work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.
  - Applies to slightly larger class of **semi-complete digraphs**.

Tournaments

- It turns out that many problems can be solved efficiently in the class of **tournaments**: work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.
  - Applies to slightly larger class of **semi-complete digraphs**.
- Elegant mathematical containment theory.

Tournaments

- It turns out that many problems can be solved efficiently in the class of **tournaments**: work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.
  - Applies to slightly larger class of **semi-complete digraphs**.
- Elegant mathematical containment theory.
- Immersion and minor orderings are WQO on semi-complete digraphs.

Tournaments

- It turns out that many problems can be solved efficiently in the class of **tournaments**: work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.
  - Applies to slightly larger class of **semi-complete digraphs**.
- Elegant mathematical containment theory.
- Immersion and minor orderings are WQO on semi-complete digraphs.
- Algorithmic applications: many FPT and XP algorithms for various versions of containment problems.

Width measures

- As in undirected graphs, width measures play crucial role.

Width measures

- As in undirected graphs, width measures play crucial role.
- **Cutwidth** corresponds to immersion order.

Width measures

- As in undirected graphs, width measures play crucial role.
- **Cutwidth** corresponds to immersion order.
  - Excluding a fixed digraph H as an immersion implies a bound on cutwidth [Chudnovsky, Fradkin, Seymour].

Width measures

- As in undirected graphs, width measures play crucial role.
- **Cutwidth** corresponds to immersion order.
  - Excluding a fixed digraph H as an immersion implies a bound on cutwidth [Chudnovsky, Fradkin, Seymour].
- **Pathwidth** corresponds to topological minor order.

Width measures

- As in undirected graphs, width measures play crucial role.
- **Cutwidth** corresponds to immersion order.
  - Excluding a fixed digraph H as an immersion implies a bound on cutwidth [Chudnovsky, Fradkin, Seymour].
- **Pathwidth** corresponds to topological minor order.
  - Excluding a fixed digraph H as a topological minor order implies a bound on pathwidth [Fradkin, Seymour].

Width measures

- As in undirected graphs, width measures play crucial role.
- **Cutwidth** corresponds to immersion order.
  - Excluding a fixed digraph H as an immersion implies a bound on cutwidth [Chudnovsky, Fradkin, Seymour].
- **Pathwidth** corresponds to topological minor order.
  - Excluding a fixed digraph H as a topological minor order implies a bound on pathwidth [Fradkin, Seymour].
- These theorems are foundations of algorithmic and WQO results.

Width measures

- As in undirected graphs, width measures play crucial role.
- **Cutwidth** corresponds to immersion order.
  - Excluding a fixed digraph H as an immersion implies a bound on cutwidth [Chudnovsky, Fradkin, Seymour].
- **Pathwidth** corresponds to topological minor order.
  - Excluding a fixed digraph H as a topological minor order implies a bound on pathwidth [Fradkin, Seymour].
- These theorems are foundations of algorithmic and WQO results.
- Approximation algorithms for both measures are hence crucial for the whole theory.

Comparison

Problem	Previous results	This work
Cutwidth approximation	$O(n^3)$ time, width $O(k^2)$	
Cutwidth exact	$O(f(k) \cdot n^3)$ time, non-uniform, non-constructive	
Pathwidth approximation	$n^{f(k)}$ time, width $O(k^2)$	
Pathwidth exact	$n^{f(k)}$ time, non-uniform, non-constructive	
Immersion	$O(f(\|H\|) \cdot n^3)$ time	
Topological containment	$n^{f(\|H\|)}$ time	

- Black: Work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.

Comparison

Problem	Previous results	This work
Cutwidth approximation	$O(n^3)$ time, width $O(k^2)$	
Cutwidth exact	$O(f(k) \cdot n^3)$ time, non-uniform, non-constructive	
Pathwidth approximation	$O(2^{O(k \log k)} \cdot n^3 \log n)$ time, width $O(k^2)$	
Pathwidth exact	$O(f(k) \cdot n^3 \log n)$ time, non-uniform, non-constructive	
Immersion	$O(f(\|H\|) \cdot n^3)$ time	
Topological containment	$O(f(\|H\|) \cdot n^3 \log n)$ time	

- Black: Work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.
- Red: Fomin and P, SODA 2013.

Comparison

Problem	Previous results	This work
Cutwidth approximation	$O(n^3)$ time, width $O(k^2)$	$O(n^2)$ time, width $O(k^2)$
Cutwidth exact	$O(f(k) \cdot n^3)$ time, non-uniform, non-constructive	$O(2^{O(k)} \cdot n^2)$ time
Pathwidth approximation	$O(2^{O(k \log k)} \cdot n^3 \log n)$ time, width $O(k^2)$	$O(kn^2)$ time, width $7k$
Pathwidth exact	$O(f(k) \cdot n^3 \log n)$ time, non-uniform, non-constructive	$O(2^{O(k \log k)} \cdot n^2)$ time
Immersion	$O(f(\lvert H\rvert) \cdot n^3)$ time	$O(2^{O(\lvert H\rvert^2 \log \lvert H\rvert)} \cdot n^2)$ time
Topological containment	$O(f(\lvert H\rvert) \cdot n^3 \log n)$ time	$O(2^{O(\lvert H\rvert \log \lvert H\rvert)} \cdot n^2)$ time

- Black: Work of Chudnovsky, Fradkin, Kim, Scott, and Seymour.
- Red: Fomin and P, SODA 2013.
- Blue: this work.

Cutwidth

- Let $T = (V, E)$ be a tournament and $\sigma = (v_1, v_2, \ldots, v_n)$ be an ordering of V.

Cutwidth

- Let $T = (V, E)$ be a tournament and $\sigma = (v_1, v_2, \ldots, v_n)$ be an ordering of V.
- We define *width* of σ to be

$$\max_{0<i<n} |E(\{v_1, v_2, \ldots, v_i\}, \{v_{i+1}, v_{i+2}, \ldots, v_n\})|$$

Cutwidth

- Let $T = (V, E)$ be a tournament and $\sigma = (v_1, v_2, \ldots, v_n)$ be an ordering of V.
- We define *width* of σ to be

$$\max_{0<i<n} |E(\{v_1, v_2, \ldots, v_i\}, \{v_{i+1}, v_{i+2}, \ldots, v_n\})|$$

- *Cutwidth* of T, denoted $\mathbf{ctw}(T)$, is minimum width among orderings of V.

Pathwidth

- A *path decomposition* $\mathcal{P}$ of $T = (V, E)$ is a sequence $[W_1, W_2, \ldots, W_p]$ of *bags* such that:

Pathwidth

- A *path decomposition* $\mathcal{P}$ of $T = (V, E)$ is a sequence $[W_1, W_2, \ldots, W_p]$ of *bags* such that:
  - $W_i \subseteq V$ and $\bigcup_{i=1}^{p} W_i = V$;

Pathwidth

- A *path decomposition* $\mathcal{P}$ of $T = (V, E)$ is a sequence $[W_1, W_2, \ldots, W_p]$ of *bags* such that:
  - $W_i \subseteq V$ and $\bigcup_{i=1}^{p} W_i = V$;
  - for every vertex v, the set of bags containing v form an interval I_v in $\mathcal{P}$;

Pathwidth

- A *path decomposition* $\mathcal{P}$ of $T = (V, E)$ is a sequence $[W_1, W_2, \ldots, W_p]$ of *bags* such that:
  - $W_i \subseteq V$ and $\bigcup_{i=1}^{p} W_i = V$;
  - for every vertex v, the set of bags containing v form an interval I_v in $\mathcal{P}$;
  - for every arc (v, w), intervals I_v and I_w either have nonempty intersection, or I_v appears later on $\mathcal{P}$ than I_w.

Pathwidth

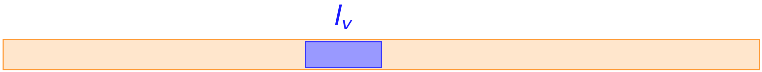
- A *path decomposition* $\mathcal{P}$ of $T = (V, E)$ is a sequence $[W_1, W_2, \ldots, W_p]$ of *bags* such that:
  - $W_i \subseteq V$ and $\bigcup_{i=1}^{p} W_i = V$;
  - for every vertex v, the set of bags containing v form an interval I_v in $\mathcal{P}$;
  - for every arc (v, w), intervals I_v and I_w either have nonempty intersection, or I_v appears later on $\mathcal{P}$ than I_w.
- Width of $\mathcal{P}$ is $\max |W_i| - 1$.

Pathwidth

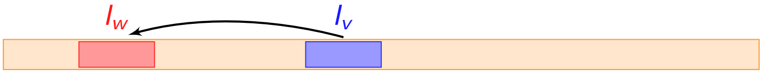
- A *path decomposition* $\mathcal{P}$ of $T = (V, E)$ is a sequence $[W_1, W_2, \ldots, W_p]$ of *bags* such that:
  - $W_i \subseteq V$ and $\bigcup_{i=1}^{p} W_i = V$;
  - for every vertex v, the set of bags containing v form an interval I_v in $\mathcal{P}$;
  - for every arc (v, w), intervals I_v and I_w either have nonempty intersection, or I_v appears later on $\mathcal{P}$ than I_w.
- Width of $\mathcal{P}$ is $\max |W_i| - 1$.
- Pathwidth of T, denoted $\mathbf{pw}(T)$, is minimum width among the path decompositions of T.

Intuition

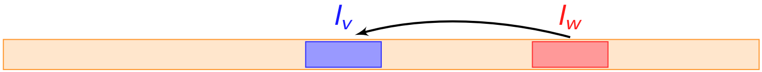
Intuition



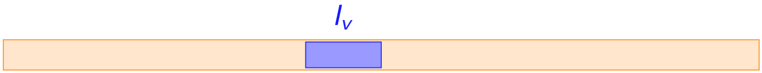
Intuition



Intuition

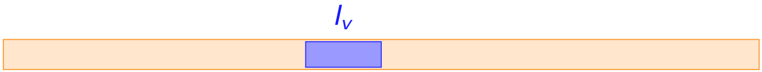


Intuition



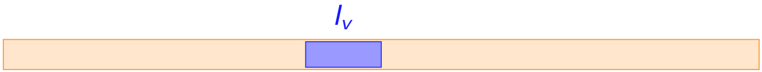
- Outdegree of a vertex approximately indicates, where the vertex should be placed in the optimal path decomposition.

Intuition



- Outdegree of a vertex approximately indicates, where the vertex should be placed in the optimal path decomposition.
  - Possible aberrations due to vertices sharing bags, but their number is limited by the width of the decomposition.

Intuition



- Outdegree of a vertex approximately indicates, where the vertex should be placed in the optimal path decomposition.
  - Possible aberrations due to vertices sharing bags, but their number is limited by the width of the decomposition.
- In particular, there should not be many vertices with similar outdegrees.

Formalizing intuition

- For cutwidth this is already enough!

Formalizing intuition

- For cutwidth this is already enough!
  - Every ordering according to outdegrees has width at most $O(\mathbf{ctw}(T)^2)$.

Formalizing intuition

- For cutwidth this is already enough!
  - Every ordering according to outdegrees has width at most $O(\mathbf{ctw}(T)^2)$.
  - Trivial approximation algorithm: just sort.

Formalizing intuition

- For cutwidth this is already enough!
    - Every ordering according to outdegrees has width at most $O(\mathbf{ctw}(T)^2)$.
    - Trivial approximation algorithm: just sort.
- For pathwidth we pursue the direction of bounding the number of vertices with similar degrees.

Main lemma

Degree tangle

(k, ℓ)-*degree tangle* is a set $X \subseteq V(T)$ such that

(i) $|X| \geqslant k$,

(ii) for every $v, w \in X$, $|d^+(v) - d^+(w)| \leqslant \ell$.

Main lemma

Degree tangle

(k, ℓ)-*degree tangle* is a set $X \subseteq V(T)$ such that

(i) $|X| \geqslant k$,

(ii) for every $v, w \in X$, $|d^+(v) - d^+(w)| \leqslant \ell$.

Main lemma

If a tournament T contains a $(5k + 2, k)$-degree tangle, then $\mathbf{pw}(T) > k$.

Main lemma

Degree tangle

(k, ℓ)-*degree tangle* is a set $X \subseteq V(T)$ such that

(i) $|X| \geqslant k$,

(ii) for every $v, w \in X$, $|d^+(v) - d^+(w)| \leqslant \ell$.

Main lemma

If a tournament T contains a $(5k+2, k)$-degree tangle, then $\mathbf{pw}(T) > k$.

- **Intuition**: The outdegrees in a low-pathwidth tournament must be really spread.

Proof of the main lemma

- Let X be a $(5k+2, k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha + k]$.

Proof of the main lemma

- Let X be a $(5k+2, k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha+k]$.
- Assume by contradiction that there is a path decomposition $\mathcal{P} = [W_1, W_2, \ldots, W_p]$ of T of width at most k.

Proof of the main lemma

- Let X be a $(5k+2, k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha + k]$.
- Assume by contradiction that there is a path decomposition $\mathcal{P} = [W_1, W_2, \ldots, W_p]$ of T of width at most k.
- By standard arguments, we may assume that $\mathcal{P}$ is a *nice path decomposition*: every bag either introduces or forgets one vertex.

Proof of the main lemma

- Let X be a $(5k+2, k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha+k]$.
- Assume by contradiction that there is a path decomposition $\mathcal{P} = [W_1, W_2, \ldots, W_p]$ of T of width at most k.
- By standard arguments, we may assume that $\mathcal{P}$ is a *nice path decomposition*: every bag either introduces or forgets one vertex.
- Hence, there is a prefix $[W_1, W_2, \ldots, W_i]$ such that $|\bigcup_{j=1}^{i} W_i| = \alpha$.

Proof of the main lemma

- Let X be a $(5k+2, k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha + k]$.
- Assume by contradiction that there is a path decomposition $\mathcal{P} = [W_1, W_2, \ldots, W_p]$ of T of width at most k.
- By standard arguments, we may assume that $\mathcal{P}$ is a *nice path decomposition*: every bag either introduces or forgets one vertex.
- Hence, there is a prefix $[W_1, W_2, \ldots, W_i]$ such that $|\bigcup_{j=1}^{i} W_i| = \alpha$.
- Let $A = \bigcup_{j=1}^{i} W_i$ and $B = \bigcup_{j=i+1}^{p} W_i$.

Proof of the main lemma

- Let X be a $(5k+2, k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha+k]$.
- Assume by contradiction that there is a path decomposition $\mathcal{P} = [W_1, W_2, \ldots, W_p]$ of T of width at most k.
- By standard arguments, we may assume that $\mathcal{P}$ is a *nice path decomposition*: every bag either introduces or forgets one vertex.
- Hence, there is a prefix $[W_1, W_2, \ldots, W_i]$ such that $|\bigcup_{j=1}^{i} W_i| = \alpha$.
- Let $A = \bigcup_{j=1}^{i} W_i$ and $B = \bigcup_{j=i+1}^{p} W_i$.
- We have that

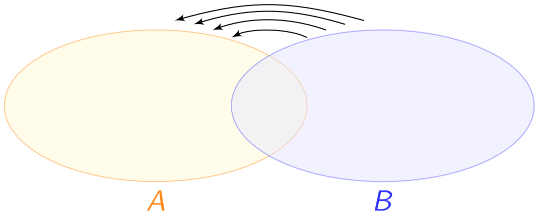
Proof of the main lemma

- Let X be a $(5k+2,k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha+k]$.
- Assume by contradiction that there is a path decomposition $\mathcal{P} = [W_1, W_2, \ldots, W_p]$ of T of width at most k.
- By standard arguments, we may assume that $\mathcal{P}$ is a *nice path decomposition*: every bag either introduces or forgets one vertex.
- Hence, there is a prefix $[W_1, W_2, \ldots, W_i]$ such that $|\bigcup_{j=1}^{i} W_i| = \alpha$.
- Let $A = \bigcup_{j=1}^{i} W_i$ and $B = \bigcup_{j=i+1}^{p} W_i$.
- We have that
    - $|A \cap B| \leqslant k$,

Proof of the main lemma

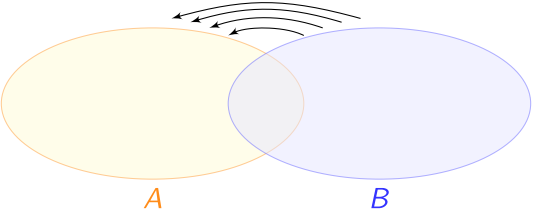
- Let X be a $(5k+2, k)$-degree tangle, and let the outdegrees in X be in the interval $[\alpha, \alpha+k]$.
- Assume by contradiction that there is a path decomposition $\mathcal{P} = [W_1, W_2, \ldots, W_p]$ of T of width at most k.
- By standard arguments, we may assume that $\mathcal{P}$ is a *nice path decomposition*: every bag either introduces or forgets one vertex.
- Hence, there is a prefix $[W_1, W_2, \ldots, W_i]$ such that $|\bigcup_{j=1}^{i} W_i| = \alpha$.
- Let $A = \bigcup_{j=1}^{i} W_i$ and $B = \bigcup_{j=i+1}^{p} W_i$.
- We have that
  - $|A \cap B| \leqslant k$,
  - and there is no arc from $A \setminus B$ to $B \setminus A$.

Proof of the main lemma



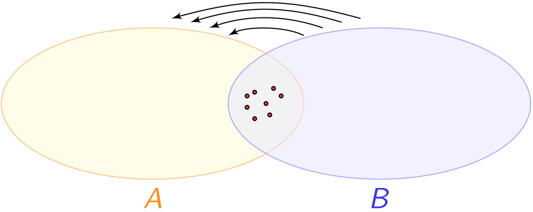
- Where vertices of X can be placed?

Proof of the main lemma



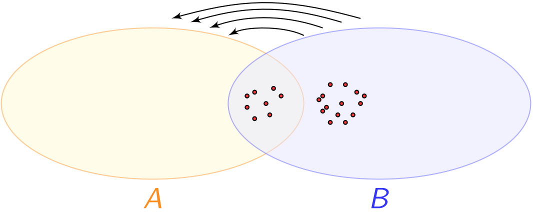
- Where vertices of X can be placed?
  - Not in $A \setminus B$, as the outdegrees there are at most $\alpha - 1$.

Proof of the main lemma



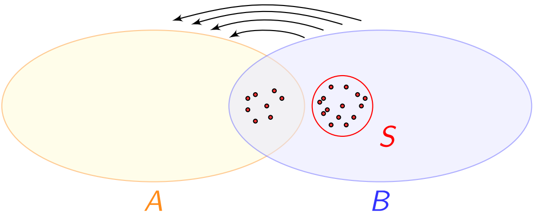
- Where vertices of X can be placed?
  - Not in $A \setminus B$, as the outdegrees there are at most $\alpha - 1$.
  - Maybe in $A \cap B$, but then only at most k of them.

Proof of the main lemma



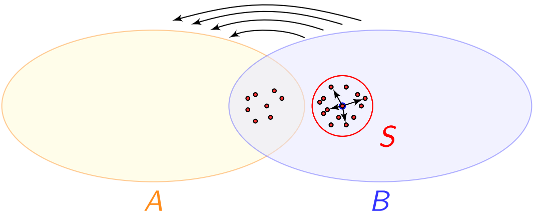
- Where vertices of X can be placed?
  - Not in $A \setminus B$, as the outdegrees there are at most $\alpha - 1$.
  - Maybe in $A \cap B$, but then only at most k of them.
  - Hence, $\geqslant 4k + 2$ vertices of X are placed in $B \setminus A$; denote them S.

Proof of the main lemma



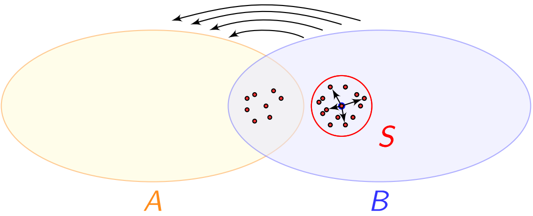
- Let's look into $T[S]$.

Proof of the main lemma



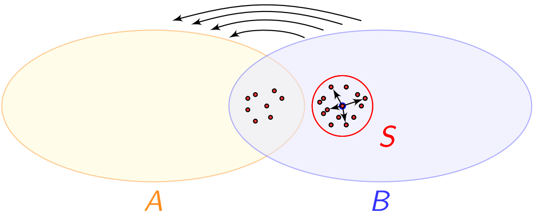
- Let's look into $T[S]$.
- As $|S| \geqslant 4k+2$, there is a vertex w of outdegree $\geqslant 2k+1$.

Proof of the main lemma



- Let's look into $T[S]$.
- As $|S| \geqslant 4k + 2$, there is a vertex w of outdegree $\geqslant 2k + 1$.
- Now $d^+(w) \geqslant |A \setminus B| + 2k + 1 \geqslant \alpha - k + 2k + 1 = \alpha + k + 1$.

Proof of the main lemma



- Let's look into $T[S]$.
- As $|S| \geqslant 4k+2$, there is a vertex w of outdegree $\geqslant 2k+1$.
- Now $d^+(w) \geqslant |A \setminus B| + 2k + 1 \geqslant \alpha - k + 2k + 1 = \alpha + k + 1$.
- Contradiction with $w \in X$.

Approximating pathwidth

- Sort the vertices according to outdegrees.

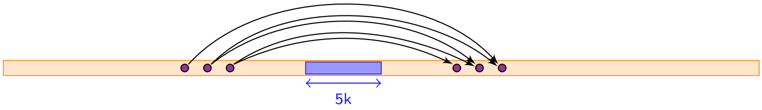
Approximating pathwidth

- Sort the vertices according to outdegrees.
- Consider a window of length $5k$ in the ordering. Any w on the right has degree at least $k+1$ larger than any v on the left.

Approximating pathwidth

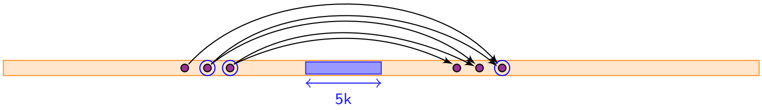
- Sort the vertices according to outdegrees.
- Consider a window of length $5k$ in the ordering. Any w on the right has degree at least $k+1$ larger than any v on the left.
- To separate left from right, we can pick:

Approximating pathwidth



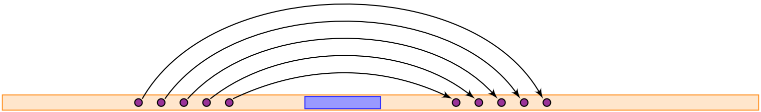
- Sort the vertices according to outdegrees.
- Consider a window of length $5k$ in the ordering. Any w on the right has degree at least $k+1$ larger than any v on the left.
- To separate left from right, we can pick:
  - greedily the whole window,

Approximating pathwidth



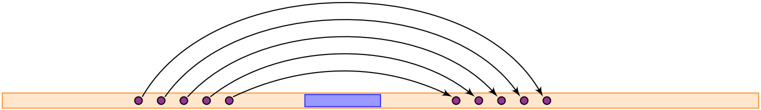
- Sort the vertices according to outdegrees.
- Consider a window of length $5k$ in the ordering. Any w on the right has degree at least $k+1$ larger than any v on the left.
- To separate left from right, we can pick:
  - greedily the whole window,
  - a vertex cover of forward arcs between left and right.

Approximating pathwidth



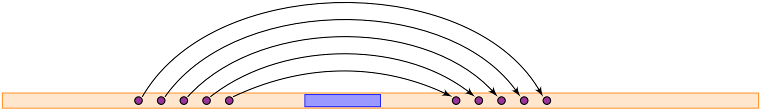
- **Observation**: A matching of size $k + 1$ of forward edges is an obstruction for pathwidth k (*matching tangle*).

Approximating pathwidth



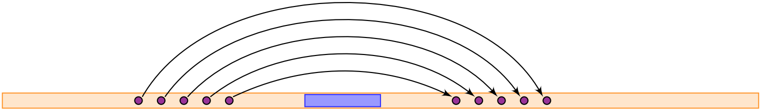
- **Observation**: A matching of size $k+1$ of forward edges is an obstruction for pathwidth k (*matching tangle*).
- Hence there is a vertex cover of size at most k.

Approximating pathwidth



- **Observation**: A matching of size $k + 1$ of forward edges is an obstruction for pathwidth k (*matching tangle*).
- Hence there is a vertex cover of size at most k.
- **Problem**: we cannot pick the vertex covers arbitrarily, as the choice must be consistent as the window slides.

Approximating pathwidth



- **Observation**: A matching of size $k+1$ of forward edges is an obstruction for pathwidth k (*matching tangle*).
- Hence there is a vertex cover of size at most k.
- **Problem**: we cannot pick the vertex covers arbitrarily, as the choice must be consistent as the window slides.
- **Solution**: use 2-approximation of vertex cover — set of all vertices that are matched in **every** maximum matching.

Exact algorithms

- **Key idea**:

Exact algorithms

- **Key idea**:
  - identify a small set of candidates for the i-th prefix of cutwidth ordering or for the i-th bag;

Exact algorithms

- **Key idea**:
  - identify a small set of candidates for the i-th prefix of cutwidth ordering or for the i-th bag;
  - run dynamic programming on the candidates.

Exact algorithms

- **Key idea**:
  - identify a small set of candidates for the *i*-th prefix of cutwidth ordering or for the *i*-th bag;
  - run dynamic programming on the candidates.
- **Cutwidth**: $2^{O(k)}$ candidates for prefix of length i.

Exact algorithms

- **Key idea**:
  - identify a small set of candidates for the i-th prefix of cutwidth ordering or for the i-th bag;
  - run dynamic programming on the candidates.
- **Cutwidth**: $2^{O(k)}$ candidates for prefix of length i.
- **Pathwidth**: roughly $\binom{O(k^2)}{k}$ candidates for each bag.

Exact algorithms

- **Key idea**:
  - identify a small set of candidates for the i-th prefix of cutwidth ordering or for the i-th bag;
  - run dynamic programming on the candidates.
- **Cutwidth**: $2^{O(k)}$ candidates for prefix of length i.
- **Pathwidth**: roughly $\binom{O(k^2)}{k}$ candidates for each bag.
  - Need to find a set where **every** sensible candidate lies.

Exact algorithms

- **Key idea**:
  - identify a small set of candidates for the i-th prefix of cutwidth ordering or for the i-th bag;
  - run dynamic programming on the candidates.
- **Cutwidth**: $2^{O(k)}$ candidates for prefix of length i.
- **Pathwidth**: roughly $\binom{O(k^2)}{k}$ candidates for each bag.
  - Need to find a set where **every** sensible candidate lies.
  - Use simple Buss kernelization instead of 2-approximation based on matchings.

Future work

Problem	This work	Future work
Cutwidth approximation	$O(n^2)$ time, width $O(k^2)$	
Cutwidth exact	$O(2^{O(k)} \cdot n^2)$ time	
Pathwidth approximation	$O(kn^2)$ time, width $7k$	
Pathwidth exact	$O(2^{O(k \log k)} \cdot n^2)$ time	
Immersion	$O(2^{O(\lvert H\rvert^2 \log \lvert H\rvert)} \cdot n^2)$ time	
Topological containment	$O(2^{O(\lvert H\rvert \log \lvert H\rvert)} \cdot n^2)$ time	

Future work

Problem	This work	Future work
Cutwidth approximation	$O(n^2)$ time, width $O(k^2)$	constant factor approximation?
Cutwidth exact	$O(2^{O(k)} \cdot n^2)$ time	
Pathwidth approximation	$O(kn^2)$ time, width $7k$	
Pathwidth exact	$O(2^{O(k \log k)} \cdot n^2)$ time	
Immersion	$O(2^{O(\lvert H\rvert^2 \log \lvert H\rvert)} \cdot n^2)$ time	
Topological containment	$O(2^{O(\lvert H\rvert \log \lvert H\rvert)} \cdot n^2)$ time	

Future work

Problem	This work	Future work
Cutwidth approximation	$O(n^2)$ time, width $O(k^2)$	constant factor approximation?
Cutwidth exact	$O(2^{O(k)} \cdot n^2)$ time	
Pathwidth approximation	$O(kn^2)$ time, width $7k$	better approximation factor?
Pathwidth exact	$O(2^{O(k \log k)} \cdot n^2)$ time	
Immersion	$O(2^{O(\|H\|^2 \log \|H\|)} \cdot n^2)$ time	
Topological containment	$O(2^{O(\|H\| \log \|H\|)} \cdot n^2)$ time	

Future work

Problem	This work	Future work
Cutwidth approximation	$O(n^2)$ time, width $O(k^2)$	constant factor approximation?
Cutwidth exact	$O(2^{O(k)} \cdot n^2)$ time	
Pathwidth approximation	$O(kn^2)$ time, width $7k$	better approximation factor?
Pathwidth exact	$O(2^{O(k \log k)} \cdot n^2)$ time	$O(2^{O(k)} \cdot n^c)$?
Immersion	$O(2^{O(\|H\|^2 \log \|H\|)} \cdot n^2)$ time	
Topological containment	$O(2^{O(\|H\| \log \|H\|)} \cdot n^2)$ time	

Future work

Problem	This work	Future work
Cutwidth approximation	$O(n^2)$ time, width $O(k^2)$	constant factor approximation?
Cutwidth exact	$O(2^{O(k)} \cdot n^2)$ time	
Pathwidth approximation	$O(kn^2)$ time, width $7k$	better approximation factor?
Pathwidth exact	$O(2^{O(k \log k)} \cdot n^2)$ time	$O(2^{O(k)} \cdot n^c)$?
Immersion	$O(2^{O(\|H\|^2 \log \|H\|)} \cdot n^2)$ time	single-exponential FPT in rooted variant?
Topological containment	$O(2^{O(\|H\| \log \|H\|)} \cdot n^2)$ time	

Future work

Problem	This work	Future work
Cutwidth approximation	$O(n^2)$ time, width $O(k^2)$	constant factor approximation?
Cutwidth exact	$O(2^{O(k)} \cdot n^2)$ time	
Pathwidth approximation	$O(kn^2)$ time, width $7k$	better approximation factor?
Pathwidth exact	$O(2^{O(k \log k)} \cdot n^2)$ time	$O(2^{O(k)} \cdot n^c)$?
Immersion	$O(2^{O(\|H\|^2 \log \|H\|)} \cdot n^2)$ time	single-exponential FPT in rooted variant?
Topological containment	$O(2^{O(\|H\| \log \|H\|)} \cdot n^2)$ time	FPT in rooted variant?

Future work

Problem	This work	Future work
Cutwidth approximation	$O(n^2)$ time, width $O(k^2)$	constant factor approximation?
Cutwidth exact	$O(2^{O(k)} \cdot n^2)$ time	$O(2^{O(\sqrt{k \log k})} \cdot n^c)$ [Fomin, P]
Pathwidth approximation	$O(kn^2)$ time, width $7k$	better approximation factor?
Pathwidth exact	$O(2^{O(k \log k)} \cdot n^2)$ time	$O(2^{O(k)} \cdot n^c)$?
Immersion	$O(2^{O(\|H\|^2 \log \|H\|)} \cdot n^2)$ time	single-exponential FPT in rooted variant?
Topological containment	$O(2^{O(\|H\| \log \|H\|)} \cdot n^2)$ time	FPT in rooted variant?

Open problems

- Constant factor approximation of cutwidth.
- Better approximation factor for pathwidth.
- Improving $2^{O(k \log k)}$ to $2^{O(k)}$ in the exact algorithm for pathwidth.
- Single-exponential algorithm for rooted immersion testing.
- FPT algorithm for vertex-disjoint paths.