Adapting the Directed Grid Theorem into an FPT algorithm

Raul Lopes

ParGO Group, Universidade Federal do Ceará, Fortaleza, Brazil

Joint work with Ana K. Maia, Ignasi Sau, and Victor Campos.
Work partially done at LIRMM, Montpellier, France.

XP and FPT

Problem with input size n, associated *parameter* k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $\mathcal{O}(n^k)$.

XP and FPT

Problem with input size n, associated *parameter* k:

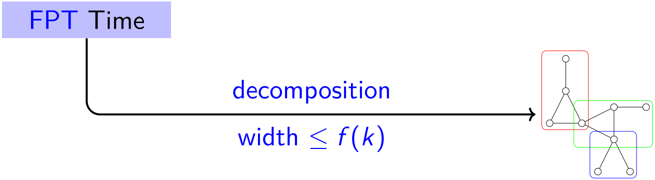
- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $\mathcal{O}(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $\mathcal{O}(2^k \cdot n^2)$ (c independent of k).

XP and FPT

Problem with input size n, associated *parameter* k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $\mathcal{O}(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $\mathcal{O}(2^k \cdot n^2)$ (c independent of k).
- $\mathrm{W}[1]$-hard problem $\Rightarrow$ strong evidence that it is *not* FPT.

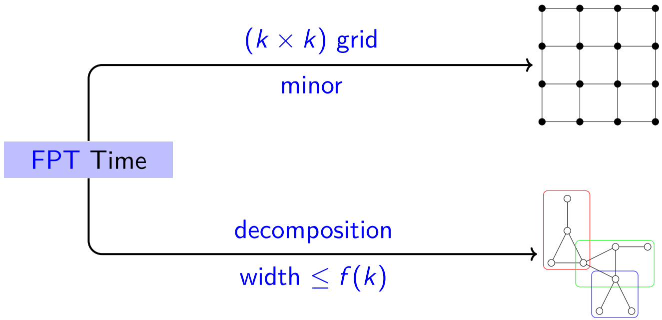
(Undirected) Grid Theorem



Grid Theorem

N. Robertson and P. Seymour.
Graph minors V. Excluding a planar graph.
Journal of Combinatorial Theory, Series B, 1986.

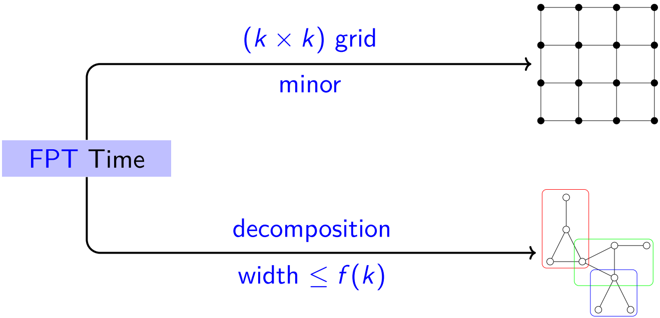
(Undirected) Grid Theorem



Grid Theorem

N. Robertson and P. Seymour.
Graph minors V. Excluding a planar graph.
Journal of Combinatorial Theory, Series B, 1986.

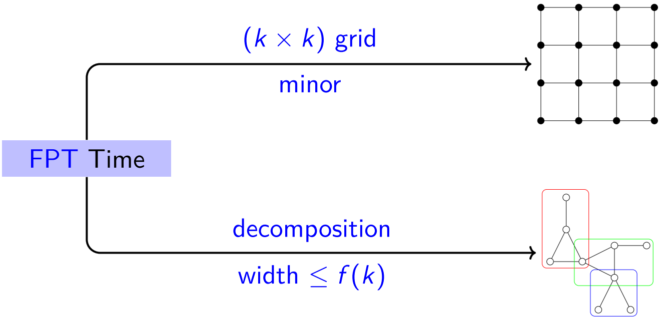
(Undirected) Grid Theorem



Applications

- Key ingredient in proof of Wagner's Conjecture.

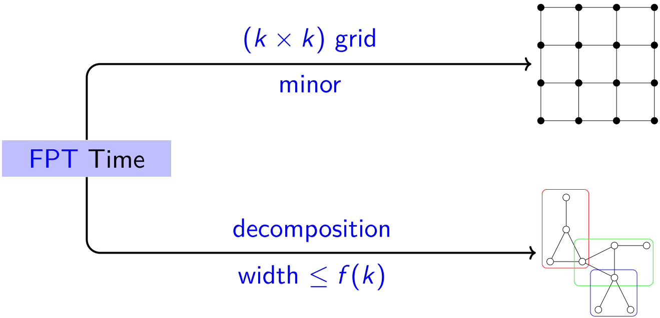
(Undirected) Grid Theorem



Applications

- Key ingredient in proof of Wagner's Conjecture.
- "Win-win" strategies:

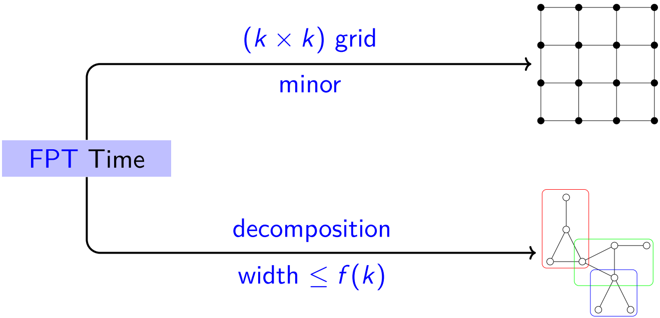
(Undirected) Grid Theorem



Applications

- Key ingredient in proof of Wagner's Conjecture.
- "Win-win" strategies:
  - Irrelevant vertex.
  - Bidimensionality.

(Undirected) Grid Theorem

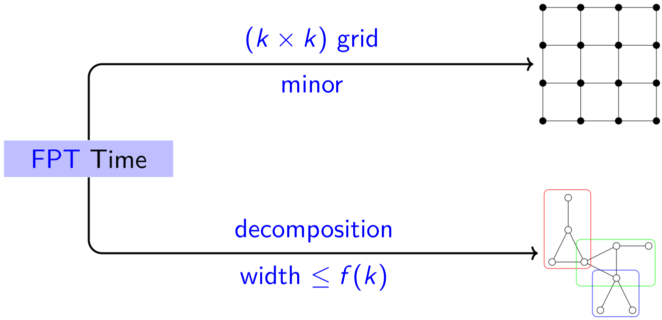


Conjecture: Directed version

Conjectured independently by

- Reed (1999).
- Johnson, Robertson, Seymour, and Thomas (2001).

(Undirected) Grid Theorem



Applications

- Key ingredient in proof of Wagner's Conjecture.
- "Win-win" strategies:
  - Irrelevant vertex.
  - Bidimensionality.

LONGEST PATH
Input: Graph G, integer k.
Question: Is there a path of size $\geq k$ in G?
$\ell p(G)$ = size of longest path in G.

LONGEST PATH

Input: Graph G, integer k.

Question: Is there a path of size $\geq k$ in G?

$\ell\mathrm{p}(G)$ = size of longest path in G.

VERTEX COVER

Input: Graph G, integer k.

Question: Is there $X \subseteq V(G)$ with $|X| \leq k$ s.t. $G \setminus X$ is edgeless?

$\mathrm{vc}(G)$ = min. size of vertex cover in G.

LONGEST PATH
Input: Graph G, integer k.
Question: Is there a path of size $\geq k$ in G?
$\ell\mathsf{p}(G)$ = size of longest path in G.

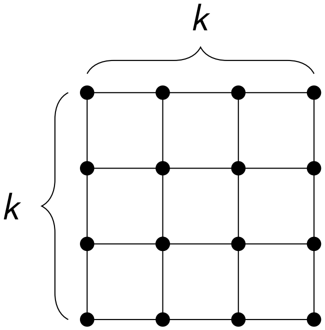
VERTEX COVER
Input: Graph G, integer k.
Question: Is there $X \subseteq V(G)$ with $|X| \leq k$ s.t. $G \setminus X$ is edgeless?
$\mathsf{vc}(G)$ = min. size of vertex cover in G.

Next part based in:

M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk and S. Saurabh.
Parameterized Algorithms.
Springer, 2015.

A look at Bidimensionality

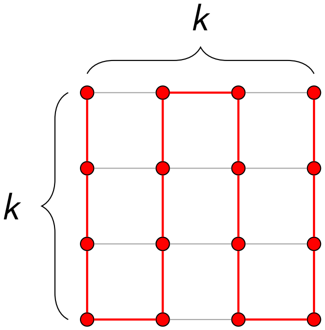
What's in a $(k \times k)$-grid?



A look at Bidimensionality

What's in a $(k \times k)$-grid?

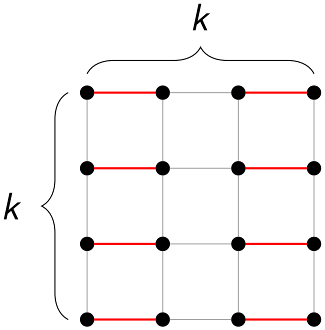
- k^2-path.



A look at Bidimensionality

What's in a $(k \times k)$-grid?

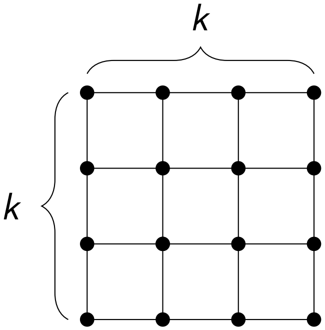
- k^2-path.
- $\left\lfloor \frac{k^2}{2} \right\rfloor$ independent edges.



A look at Bidimensionality

What's in a $(k \times k)$-grid?

- k^2-path.
- $\left\lfloor \frac{k^2}{2} \right\rfloor$ independent edges.

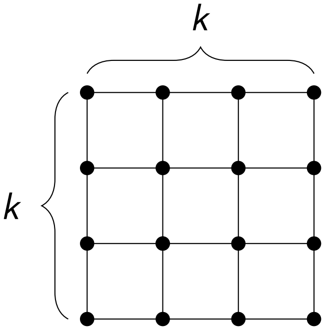


FACT: Longest Path and Vertex Cover are *minor closed*.

A look at Bidimensionality

What's in a $(k \times k)$-grid?

- k^2-path.
- $\left\lfloor \frac{k^2}{2} \right\rfloor$ independent edges.

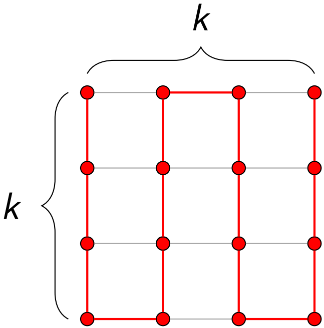


FACT: LONGEST PATH and VERTEX COVER are *minor closed*.

A look at Bidimensionality

What's in a $(k \times k)$-grid?

- k^2-path.
- $\left\lfloor \frac{k^2}{2} \right\rfloor$ independent edges.



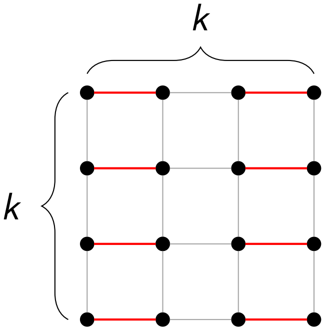
FACT: LONGEST PATH and VERTEX COVER are *minor closed*.

$$G \text{ has } (\sqrt{k} \times \sqrt{k})\text{-grid minor} \implies \begin{cases} \ell p(G) \geq k. \\ \\ \end{cases}$$

A look at Bidimensionality

What's in a $(k \times k)$-grid?

- k^2-path.
- $\left\lfloor \frac{k^2}{2} \right\rfloor$ independent edges.



FACT: LONGEST PATH and VERTEX COVER are *minor closed*.

$$G \text{ has } (\sqrt{k} \times \sqrt{k})\text{-grid minor} \implies \begin{cases} \ell p(G) \geq k. \\ vc(G) \geq k/2. \end{cases}$$

Theorem (Planar excluded grid theorem)

Planar G + $\mathrm{tw}(G) \geq 9t/2$ *then G contains $(t \times t)$-grid minor.*
$\triangleright$ *Grid or decomposition found in $\mathcal{O}(n^2)$ time.*

N. Robertson, P. Seymour, and R. Thomas.
Quickly excluding a planar graph.
Journal of Combinatorial Theory, Series B, 1994.

Q-P. Gu and H. Tamaki.
Improved Bounds on the planar branchwidth with respect to the largest grid minor size
Algorithmica, 2012.

Theorem (Planar excluded grid theorem)

Planar G + $\text{tw}(G) \geq 9t/2$ *then* G *contains* $(t \times t)$*-grid minor.*
▷ *Grid or decomposition found in* $\mathcal{O}(n^2)$ *time.*

$$+$$

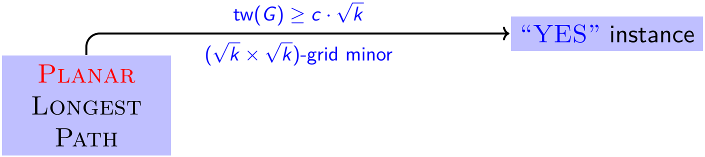
G has $(\sqrt{k} \times \sqrt{k})$-grid minor $\implies \ell\text{p}(G) \geq k$.

Theorem (Planar excluded grid theorem)

Planar G + $\mathrm{tw}(G) \geq 9t/2$ then G contains $(t \times t)$-grid minor.
$\triangleright$ *Grid or decomposition found in $\mathcal{O}(n^2)$ time.*

$$+$$

$$G \text{ has } (\sqrt{k} \times \sqrt{k})\text{-grid minor} \implies \ell\mathrm{p}(G) \geq k.$$

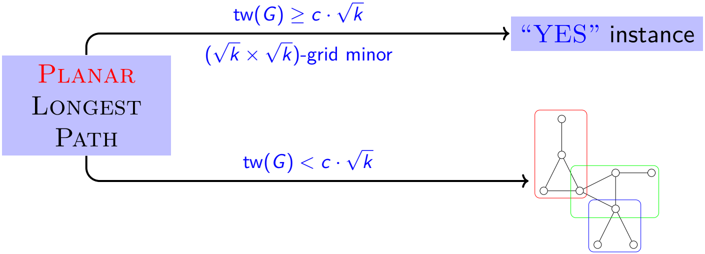


Theorem (Planar excluded grid theorem)

Planar G + $\text{tw}(G) \geq 9t/2$ *then G contains $(t \times t)$-grid minor.*
▷ *Grid or decomposition found in $\mathcal{O}(n^2)$ time.*

$$+$$

$$G \text{ has } (\sqrt{k} \times \sqrt{k})\text{-grid minor} \implies \ell\text{p}(G) \geq k.$$



Theorem (Planar excluded grid theorem)

Planar G + $\text{tw}(G) \geq 9t/2$ then G contains $(t \times t)$-grid minor.
▷ *Grid or decomposition found in $\mathcal{O}(n^2)$ time.*

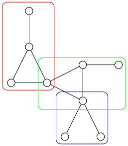
$$+$$

G has $(\sqrt{k} \times \sqrt{k})$-grid minor $\implies \ell\text{p}(G) \geq k$.

PLANAR LONGEST PATH

$\text{tw}(G) \geq c \cdot \sqrt{k}$ — $(\sqrt{k} \times \sqrt{k})$-grid minor → "YES" instance

$\text{tw}(G) < c \cdot \sqrt{k}$ →



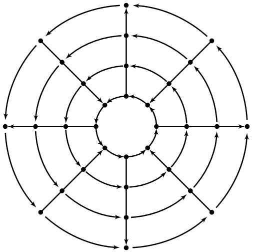
LONGEST PATH in $2^{\mathcal{O}(\text{tw})} \cdot n$ time → PLANAR LONGEST PATH in $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ time

- Subexponential algorithms for planar Steiner Tree, Feedback Vertex Set, Longes Path, Vertex Cover, Dominating Set...

- Subexponential algorithms for planar Steiner Tree, Feedback Vertex Set, Longes Path, Vertex Cover, Dominating Set...
- Similar framework for digraphs?

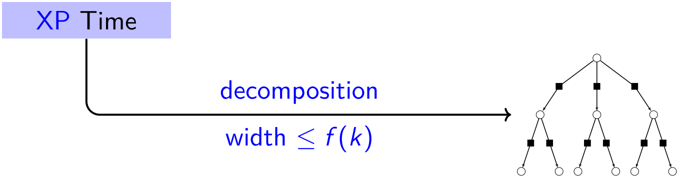
- Subexponential algorithms for planar Steiner Tree, Feedback Vertex Set, Longes Path, Vertex Cover, Dominating Set...
- Similar framework for digraphs?
- Start with a (kind of) grid and a decomposition.

- Subexponential algorithms for planar Steiner Tree, Feedback Vertex Set, Longes Path, Vertex Cover, Dominating Set...
- Similar framework for digraphs?
- Start with a (kind of) grid and a decomposition.



Cylindrical grid of order 4.

Directed Grid Theorem



Proof - 16 years later

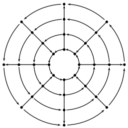
K. Kawarabayashi and S. Kreutzer.
The Directed Grid Theorem
STOC, 2015

Directed Grid Theorem

XP Time

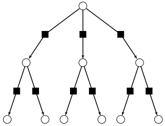
order k

cylindrical grid minor



decomposition

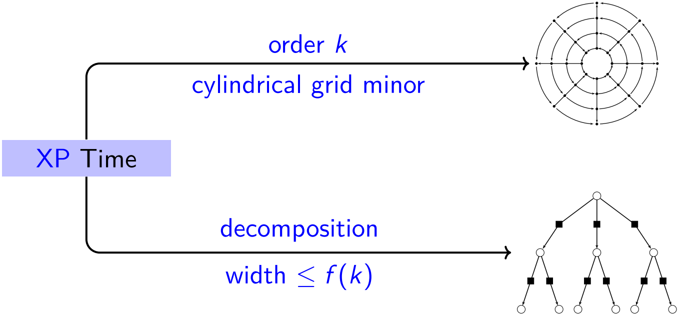
width $\leq f(k)$



Proof - 16 years later

K. Kawarabayashi and S. Kreutzer.
The Directed Grid Theorem
STOC, 2015

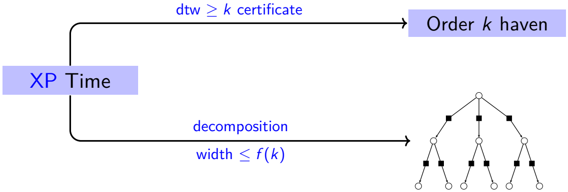
Directed Grid Theorem



Our result

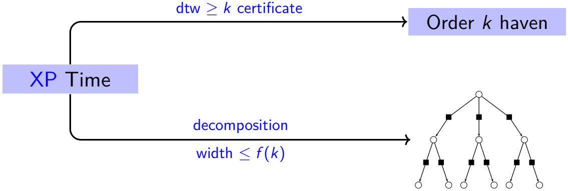
$$\text{XP} \rightarrow \text{FPT}$$

Understanding the Directed Grid Theorem



T. Johnson, N. Robertson, P. Seymour and R. Thomas.
Directed tree-width
JCTB, 2001

Understanding the Directed Grid Theorem

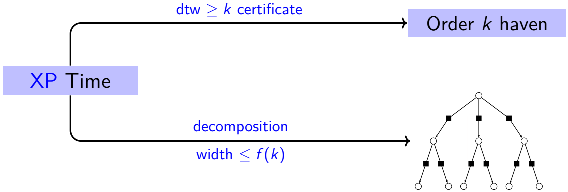


- Large haven ⇒ Large bramble.

B. Reed.
Introducing directed tree-width
ENDM, 1999

Kawarabayashi and Kreutzer.
The Directed Grid Theorem
STOC, 2015

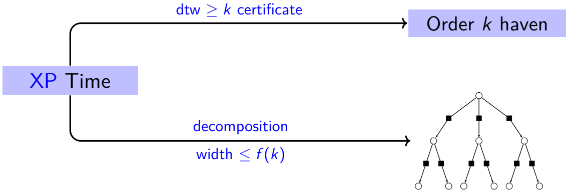
Understanding the Directed Grid Theorem



- Large haven ⇒ Large bramble.
- Large bramble ⇒ Well linked long path.

K. Kawarabayashi and S. Kreutzer.
The Directed Grid Theorem
STOC, 2015

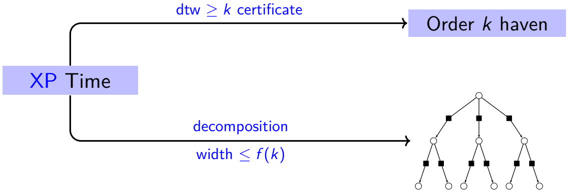
Understanding the Directed Grid Theorem



- Large haven $\Rightarrow$ Large bramble.
- Large bramble $\Rightarrow$ Well linked long path.
- Well linked long path $\Rightarrow$ Cylindrical Grid.

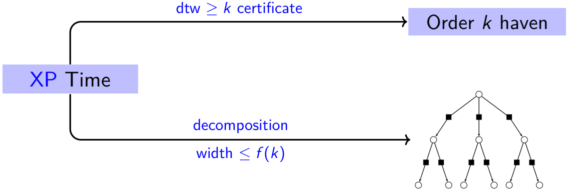
K. Kawarabayashi and S. Kreutzer.
The Directed Grid Theorem
STOC, 2015

Understanding the Directed Grid Theorem



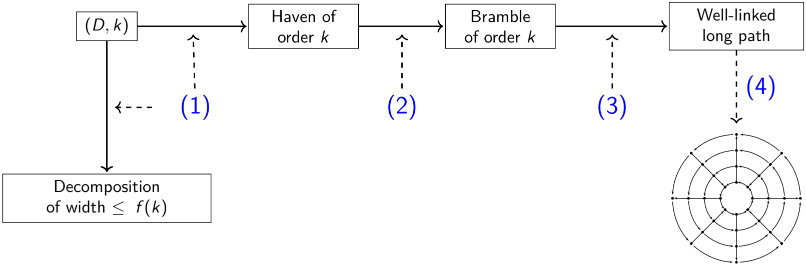
- Large haven $\Rightarrow$ Large bramble.
- Large bramble $\Rightarrow$ Well linked long path.
- Well linked long path $\Rightarrow$ Cylindrical Grid.
- Constructive proof yields an XP algorithm.

Understanding the Directed Grid Theorem



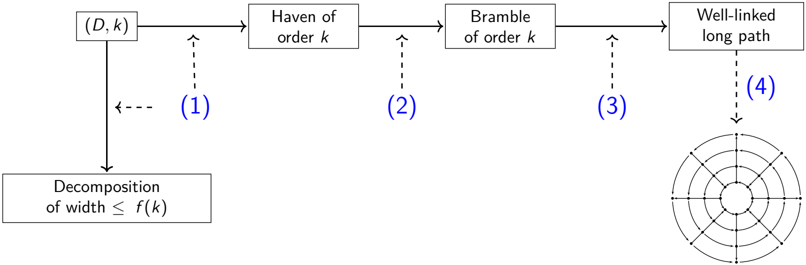
- Large haven ⇒ Large bramble.
- Large bramble ⇒ Well linked long path.
- Well linked long path ⇒ Cylindrical Grid.
- Constructive proof yields an XP algorithm.
- Analyze what needs to change to achieve FPT time.

Directed Grid Theorem: constructive proof (I)



(1) Haven vs Decomposition in XP time.

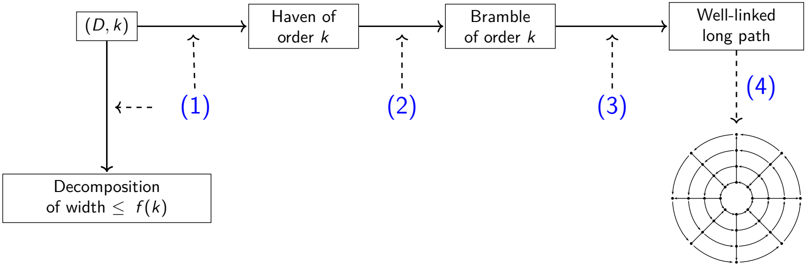
Directed Grid Theorem: constructive proof (*I*)



(1) Haven vs Decomposition in XP time.

(2) Haven ⇒ Bramble of *size* $n^{\mathcal{O}(k)}$.

Directed Grid Theorem: constructive proof (I)

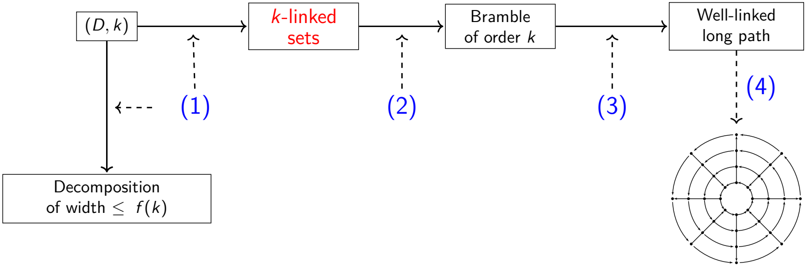


(1) Haven vs Decomposition in XP time.

(2) Haven ⇒ Bramble of *size* $n^{\mathcal{O}(k)}$.

(3) Bramble ⇒ Well-linked long path (working with hitting sets XP time).

Directed Grid Theorem: constructive proof (*I*)



(1) Haven vs Decomposition in XP time.

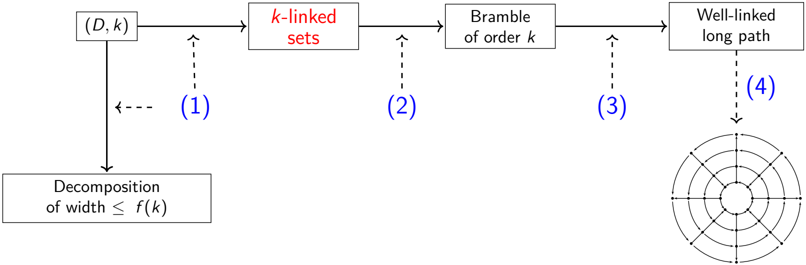
(2) Haven $\Rightarrow$ Bramble of *size* $n^{\mathcal{O}(k)}$.

(3) Bramble $\Rightarrow$ Well-linked long path (working with hitting sets XP time).

---

(1) *k-linked sets* vs Decomposition in FPT time.

Directed Grid Theorem: constructive proof (I)



(1) Haven vs Decomposition in XP time.

(2) Haven $\Rightarrow$ Bramble of *size* $n^{\mathcal{O}(k)}$.

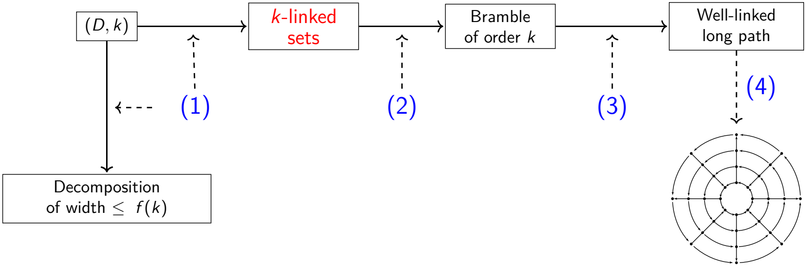
(3) Bramble $\Rightarrow$ Well-linked long path (working with hitting sets XP time).

---

(1) *k-linked sets* vs Decomposition in FPT time.

(2) *k-linked sets* $\Rightarrow$ Bramble that is *easier* to work with.

Directed Grid Theorem: constructive proof (*I*)



(1) Haven vs Decomposition in XP time.

(2) Haven ⇒ Bramble of *size* $n^{\mathcal{O}(k)}$.

(3) Bramble ⇒ Well-linked long path (working with hitting sets XP time).

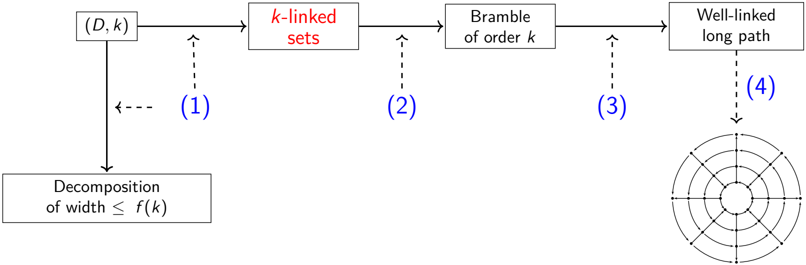
---

(1) *k-linked sets* vs Decomposition in FPT time.

(2) *k-linked sets* ⇒ Bramble that is *easier* to work with.

(3) Bramble ⇒ Well-linked long path (working with hitting sets in FPT time).

Directed Grid Theorem: constructive proof (I)



(1) Haven vs Decomposition in XP time.

(2) Haven $\Rightarrow$ Bramble of *size* $n^{\mathcal{O}(k)}$.

(3) Bramble $\Rightarrow$ Well-linked long path (working with hitting sets XP time).

---

(1) *k-linked sets* vs Decomposition in FPT time.

(2) *k-linked sets* $\Rightarrow$ Bramble that is *easier* to work with.

(3) Bramble $\Rightarrow$ Well-linked long path (working with hitting sets in FPT time).

Arboreal decompositions: J.R.S.T. algorithm

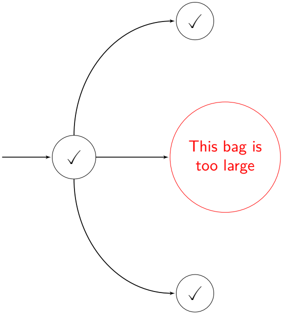
1. Starts with trivial decomposition: $V(D)$ in one bag.

Arboreal decompositions: J.R.S.T. algorithm

1. Starts with trivial decomposition: $V(D)$ in one bag.
2. ∃? bag that is "*too large*"?

   NO: End.

   YES: (Try to) break it.

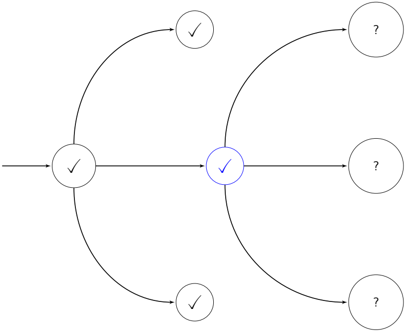


Arboreal decompositions: J.R.S.T. algorithm

1. Starts with trivial decomposition: $V(D)$ in one bag.
2. $\exists$? bag that is "*too large*"?

   NO: End.

   YES: (Try to) break it.



Large bags and balanced separators

- Given $T \subseteq V(D)$:

Definition (Balanced Separators, k-linked sets)

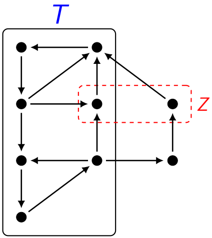
▷ $Z \subseteq V(D)$ is a *T-balanced separator* if $|T \cap V(C)| \leq \left\lfloor \frac{|T|}{2} \right\rfloor$ for every **strong** component C of $D \setminus Z$.

Large bags and balanced separators

- Given $T \subseteq V(D)$:

Definition (Balanced Separators, k-linked sets)

▷ $Z \subseteq V(D)$ is a *T-balanced separator* if $|T \cap V(C)| \leq \left\lfloor \frac{|T|}{2} \right\rfloor$ for every **strong** component C of $D \setminus Z$.



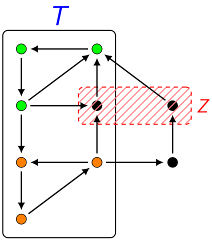
- $|T| = 7$, $\left\lfloor \frac{|T|}{2} \right\rfloor = 3$.

Large bags and balanced separators

- Given $T \subseteq V(D)$:

Definition (Balanced Separators, k-linked sets)

▷ $Z \subseteq V(D)$ is a *T-balanced separator* if $|T \cap V(C)| \leq \left\lfloor \frac{|T|}{2} \right\rfloor$ for every **strong** component C of $D \setminus Z$.



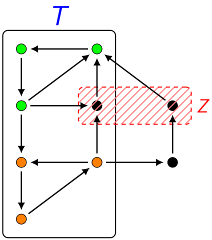
- $|T| = 7$, $\left\lfloor \frac{|T|}{2} \right\rfloor = 3$.
- Z is T-balanced Separator.

Large bags and balanced separators

- Given $T \subseteq V(D)$:

Definition (Balanced Separators, k-linked sets)

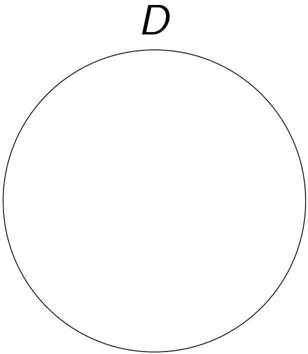
- ▷ $Z \subseteq V(D)$ is a *T-balanced separator* if $|T \cap V(C)| \leq \left\lfloor \frac{|T|}{2} \right\rfloor$ for every **strong** component C of $D \setminus Z$.
- ▷ T is *k-linked* if every T-balanced separator has *size* $\geq k+1$.



- $|T| = 7$, $\left\lfloor \frac{|T|}{2} \right\rfloor = 3$.
- Z is T-balanced Separator.

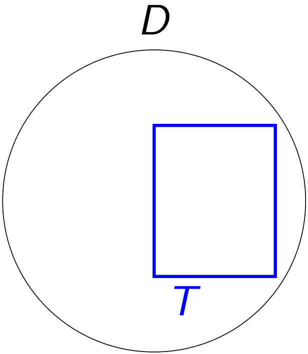
Key ingredient: $T \subseteq$ Large bag X, $|T| \leq 2k-1$: is T $(k-1)$-linked?

Finding balanced separators: XP part of JRST algorithm



On each iteration:

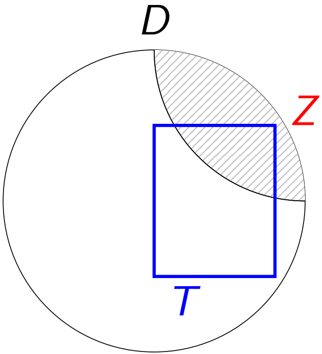
Finding balanced separators: XP part of JRST algorithm



On each iteration:

- Given $|T| \leq 2k - 1$.

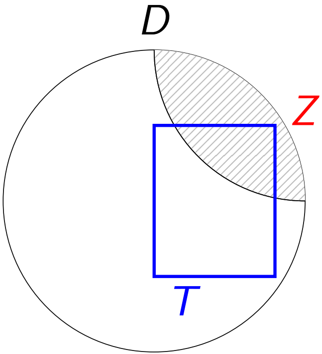
Finding balanced separators: XP part of JRST algorithm



On each iteration:

- Given $|T| \leq 2k - 1$.
- Find $Z \subseteq V(D)$:

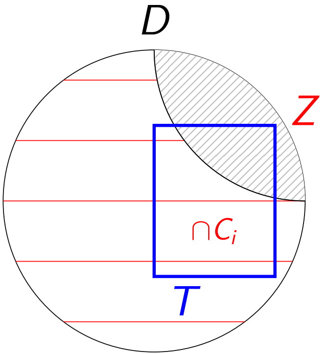
Finding balanced separators: XP part of JRST algorithm



On each iteration:

- Given $|T| \leq 2k - 1$.
- Find $Z \subseteq V(D)$:
  - $|Z| \leq k - 1$; and

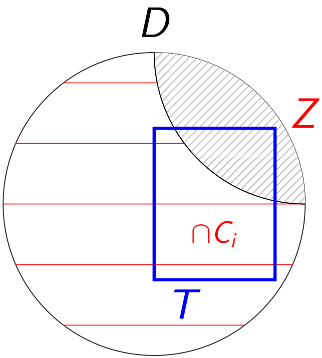
Finding balanced separators: XP part of JRST algorithm



On each iteration:

- Given $|T| \leq 2k - 1$.
- Find $Z \subseteq V(D)$:
  - $|Z| \leq k - 1$; and
  - $|C \cap T| \leq k - 1$ for strong components of $D \setminus Z$.

Finding balanced separators: XP part of JRST algorithm

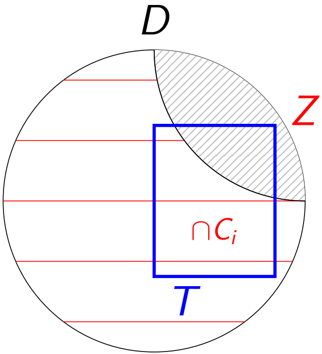


On each iteration:

- Given $|T| \leq 2k-1$.
- Find $Z \subseteq V(D)$:
  - $|Z| \leq k-1$; and
  - $|C \cap T| \leq k-1$ for strong components of $D \setminus Z$.

$\mathcal{O}(n^k)$ algorithm:

Finding balanced separators: XP part of JRST algorithm



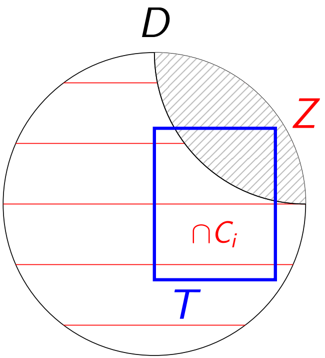
On each iteration:

- Given $|T| \leq 2k - 1$.
- Find $Z \subseteq V(D)$:
  - $|Z| \leq k - 1$; and
  - $|C \cap T| \leq k - 1$ for strong components of $D \setminus Z$.

$\mathcal{O}(n^k)$ algorithm:

▷ Enumerate strong components of $D \setminus Z$ for each of the $\binom{n}{k-1}$ sets Z.

Finding balanced separators: XP part of JRST algorithm



On each iteration:

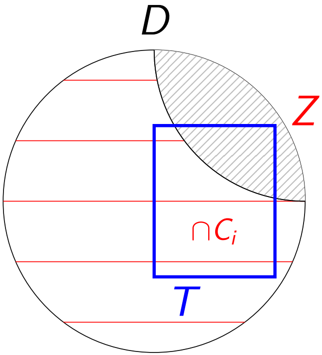
- Given $|T| \leq 2k - 1$.
- Find $Z \subseteq V(D)$:
  - $|Z| \leq k - 1$; and
  - $|C \cap T| \leq k - 1$ for strong components of $D \setminus Z$.

$\mathcal{O}(n^k)$ algorithm:

▷ Enumerate strong components of $D \setminus Z$ for each of the $\binom{n}{k-1}$ sets Z.

- **Every test is positive $\Longrightarrow$ decomposition.**

Finding balanced separators: XP part of JRST algorithm



On each iteration:

- Given $|T| \leq 2k - 1$.
- Find $Z \subseteq V(D)$:
  - $|Z| \leq k - 1$; and
  - $|C \cap T| \leq k - 1$ for strong components of $D \setminus Z$.

$\mathcal{O}(n^k)$ algorithm:

- Enumerate strong components of $D \setminus Z$ for each of the $\binom{n}{k-1}$ sets Z.

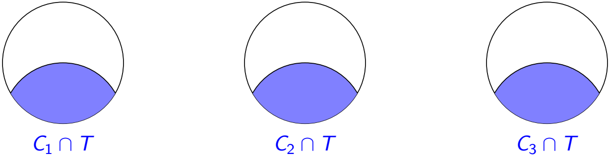
- **Every test is positive $\implies$ decomposition.**
- **Any negative $\implies$ $(k-1)$-linked set T.**

Topological order and balanced separators

- Z is T-balanced separator $\implies$

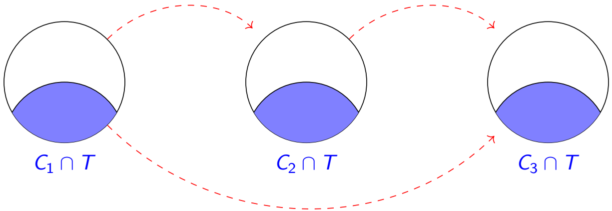
Topological order and balanced separators

- Z is T-balanced separator $\implies$
  - Each **strong** component of $D \setminus Z$ contains at most $k - 1$ vertices of T.



Topological order and balanced separators

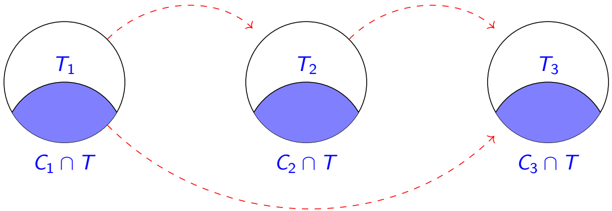
- Z is T-balanced separator $\implies$
  - Each **strong** component of $D \setminus Z$ contains at most $k-1$ vertices of T.
- (Reverse) *topological order* of **strong** components with $C_i \cap T \neq \emptyset$.



No $C_i \to C_j$ path when $i > j$

Topological order and balanced separators

- Z is T-balanced separator $\implies$
  - Each **strong** component of $D \setminus Z$ contains at most $k-1$ vertices of T.
- (Reverse) *topological order* of **strong** components with $C_i \cap T \neq \emptyset$.



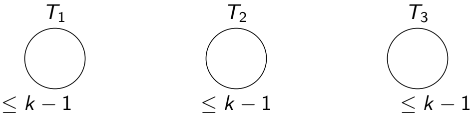
No $C_i \to C_j$ path when $i > j$ $\implies$ No $T_i \to T_j$ path when $i > j$

Similar problem $\mathcal{P}$

- Given:
  - $|T| \leq 2k - 1$.

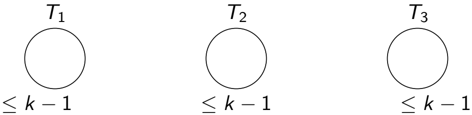
Similar problem $\mathcal{P}$

- Given:
  - $|T| \leq 2k - 1$.
  - Ordered partition $T_1, \ldots, T_r$ of T with $|T_i| \leq k - 1$.



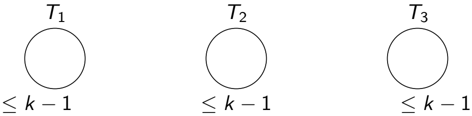
Similar problem $\mathcal{P}$

- Given:
  - $|T| \leq 2k - 1$.
  - Ordered partition $T_1, \ldots, T_r$ of T with $|T_i| \leq k - 1$.
- Find: $Z \subseteq V(D)$ such that:



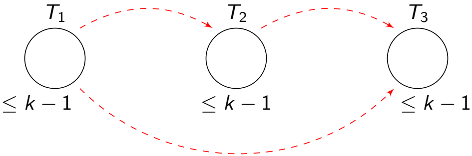
Similar problem $\mathcal{P}$

- Given:
  - $|T| \leq 2k - 1$.
  - Ordered partition $T_1, \ldots, T_r$ of T with $|T_i| \leq k - 1$.
- Find: $Z \subseteq V(D)$ such that:
  1. $|Z| \leq k - 1$; and



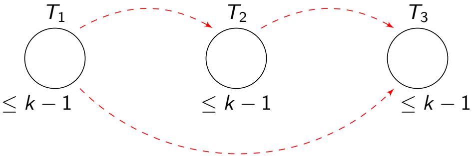
Similar problem $\mathcal{P}$

- Given:
  - $|T| \leq 2k - 1$.
  - Ordered partition $T_1, \ldots, T_r$ of T with $|T_i| \leq k - 1$.
- Find: $Z \subseteq V(D)$ such that:
  1. $|Z| \leq k - 1$; and
  2. there are **no** $T_i \to T_j$ paths for $i < j$ in $D \setminus Z$.



Similar problem $\mathcal{P}$

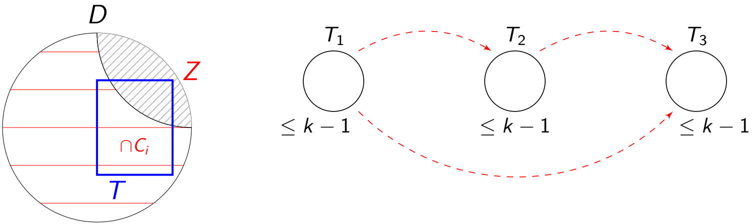
- **Given:**
  - $|T| \leq 2k - 1$.
  - Ordered partition $T_1, \ldots, T_r$ of T with $|T_i| \leq k - 1$.
- **Find:** $Z \subseteq V(D)$ such that:
  1. $|Z| \leq k - 1$; and
  2. there are **no** $T_i \to T_j$ paths for $i < j$ in $D \setminus Z$.



Reduces to FPT problem

R. Erbacher, T. Jaeger, N. Talele and J. Teutsch
Directed Multicut with Linearly Ordered Terminals
CoRR abs/1407.7498, 2014.

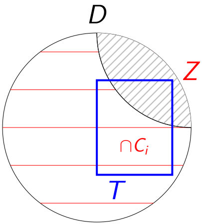
Balanced separators and $\mathcal{P}$

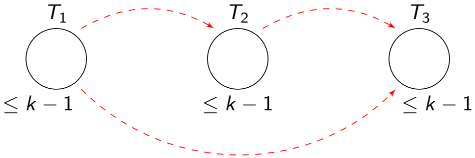


Lemma

∃ bal. separator $|Z| \leq k-1 \Leftrightarrow \mathcal{P}$ *positive for some partition* $T_1, \ldots, T_r$ *of* $T \setminus Z$.

Balanced separators and $\mathcal{P}$



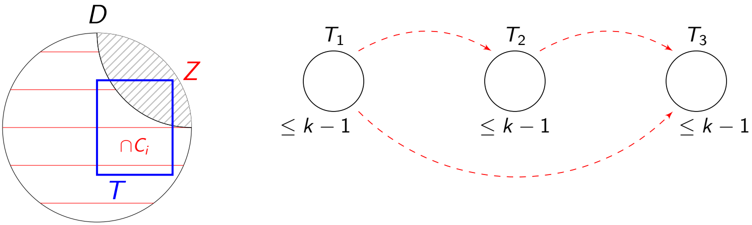


Lemma

$\exists$ *bal. separator* $|Z| \leq k-1 \Leftrightarrow \mathcal{P}$ *positive for some partition* $T_1, \ldots, T_r$ *of* $T \setminus Z$.

- #Ordered partitions of T is $2^{\mathcal{O}(k \log k)}$ (k-th ordered Bell number, $|T| \leq 2k-1$).

Balanced separators and $\mathcal{P}$

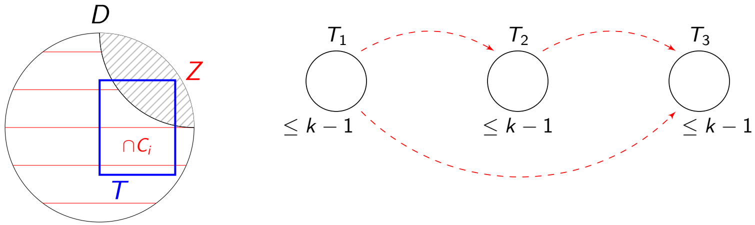


Lemma

$\exists$ bal. separator $|Z| \leq k-1 \Leftrightarrow \mathcal{P}$ positive for some partition $T_1, \ldots, T_r$ of $T \setminus Z$.

- #Ordered partitions of T is $2^{\mathcal{O}(k \log k)}$ (k-th ordered Bell number, $|T| \leq 2k-1$).
- Find balanced separators in FPT time w. param. k.

Balanced separators and $\mathcal{P}$

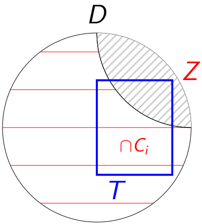


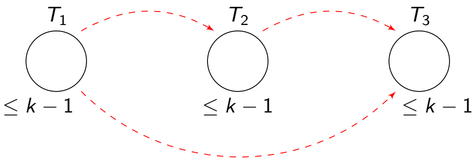
Lemma

$\exists$ bal. separator $|Z| \leq k-1 \Leftrightarrow \mathcal{P}$ positive for some partition $T_1, \ldots, T_r$ of $T \setminus Z$.

- #Ordered partitions of T is $2^{\mathcal{O}(k \log k)}$ (k-th ordered Bell number, $|T| \leq 2k-1$).
- Find balanced separators in FPT time w. param. k.
- k-linked sets vs Decomposition in FPT time.

Balanced separators and $\mathcal{P}$



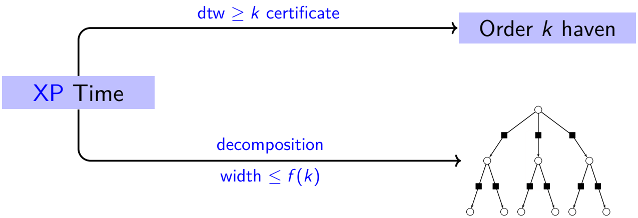


Lemma

$\exists$ bal. separator $|Z| \leq k-1 \Leftrightarrow \mathcal{P}$ positive for some partition $T_1, \ldots, T_r$ of $T \setminus Z$.

- #Ordered partitions of T is $2^{\mathcal{O}(k \log k)}$ (k-th ordered Bell number, $|T| \leq 2k-1$).
- Find balanced separators in FPT time w. param. k.
- k-linked sets vs Decomposition in FPT time.
- We solve more general version, which we named "*Partitioning sets*".

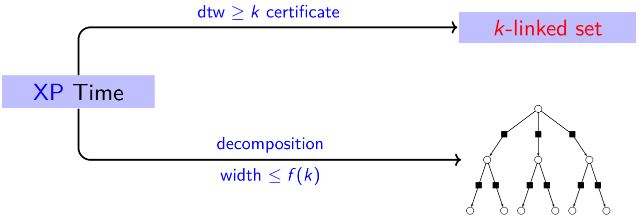
k-linked sets vs Decompositions



Theorem (Johnson et al.'01)

In XP time:
produce arboreal decomposition of width $\leq 3k - 2$ *or haven of order* k

k-linked sets vs Decompositions

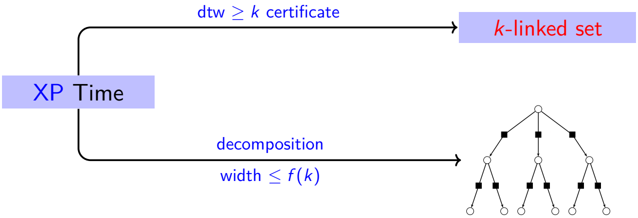


Theorem (Johnson et al.'01)

In XP time:
produce arboreal decomposition of width $\leq 3k-2$ or $(k-1)$-linked set T.

- Each iteration decides if a given set T is $(k-1)$-linked.

k-linked sets vs Decompositions

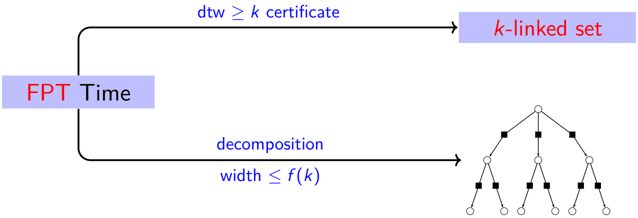


Theorem (Johnson et al.'01)

In XP time:
produce arboreal decomposition of width $\leq 3k - 2$ or $(k-1)$-linked set T.

- Each iteration decides if a given set T is $(k-1)$-linked.
- Only step that is (originally) done in XP time.

k-linked sets vs Decompositions

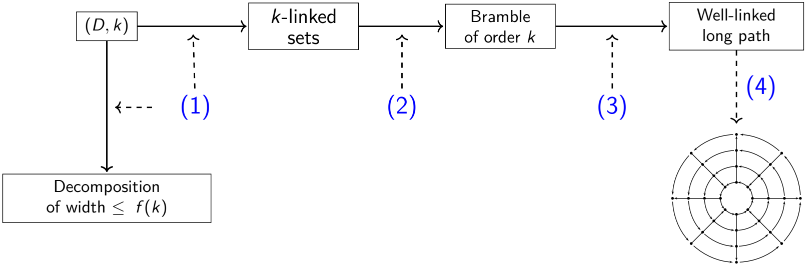


Theorem (Johnson et al.'01)

In XP time:
produce arboreal decomposition of width $\leq 3k-2$ or $(k-1)$-linked set T.

- Each iteration decides if a given set T is $(k-1)$-linked.
- Only step that is (originally) done in XP time.
- We solve it in FPT time.

Directed Grid theorem: constructive proof (*II*)



(1) Haven vs Decomposition in XP time.

(2) Haven $\Rightarrow$ Bramble of *size* $n^{\mathcal{O}(k)}$.

(3) Bramble $\Rightarrow$ Well-linked long path: working with hitting sets (XP time).

✓(1) *k-linked sets* vs Decomposition in FPT time.

(2) *k-linked sets* $\Rightarrow$ Bramble that is *easier* to work with.

(3) Bramble $\Rightarrow$ Well-linked long path: working with hitting sets (FPT time).

Brambles in digraphs

Definition (Brambles on digraphs)

- Family of strongly connected subgraphs $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ s.t.

Brambles in digraphs

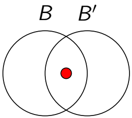
Definition (Brambles on digraphs)

- Family of strongly connected subgraphs $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ s.t.
- if $\{B, B'\} \subseteq \mathcal{B}$ then either

Brambles in digraphs

Definition (Brambles on digraphs)

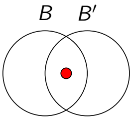
- Family of strongly connected subgraphs $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ s.t.
- if $\{B, B'\} \subseteq \mathcal{B}$ then either
  1. $V(B) \cap V(B') \neq \emptyset$ or

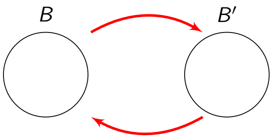


Brambles in digraphs

Definition (Brambles on digraphs)

- Family of strongly connected subgraphs $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ s.t.
- if $\{B, B'\} \subseteq \mathcal{B}$ then either
  1. $V(B) \cap V(B') \neq \emptyset$ or
  2. edges from $V(B)$ to $V(B')$ and from $V(B')$ to $V(B)$.

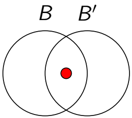


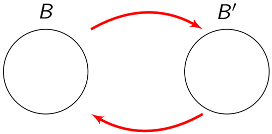


Brambles in digraphs

Definition (Brambles on digraphs)

- Family of strongly connected subgraphs $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ s.t.
- if $\{B, B'\} \subseteq \mathcal{B}$ then either
  1. $V(B) \cap V(B') \neq \emptyset$ or
  2. edges from $V(B)$ to $V(B')$ and from $V(B')$ to $V(B)$.



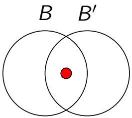


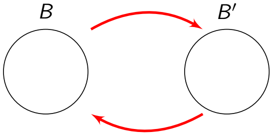
- hitting set of $\mathcal{B}$ = set of vertices touching every $B \in \mathcal{B}$.

Brambles in digraphs

Definition (Brambles on digraphs)

- Family of strongly connected subgraphs $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ s.t.
- if $\{B, B'\} \subseteq \mathcal{B}$ then either
  1. $V(B) \cap V(B') \neq \emptyset$ or
  2. edges from $V(B)$ to $V(B')$ and from $V(B')$ to $V(B)$.





- hitting set of $\mathcal{B}$ = set of vertices touching every $B \in \mathcal{B}$.
- order of $\mathcal{B}$ = minimum size of hitting set.

Working with brambles

Bramble $\mathcal{B} = \{B_1, B_2, \ldots, B_\ell\}$.

- Naive approach to find hitting sets w. size k (assuming $|\mathcal{B}| > k$):

Working with brambles

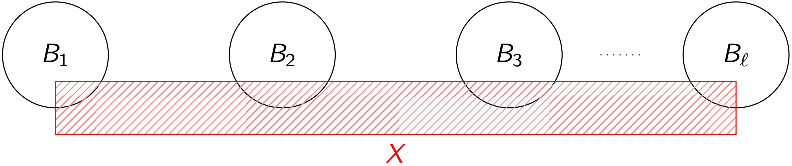
Bramble $\mathcal{B} = \{B_1, B_2, \ldots, B_\ell\}$.

- Naive approach to find hitting sets w. size k (assuming $|\mathcal{B}| > k$):
  - for each $X \subseteq V(D)$ w. $|X| \leq k$,

Working with brambles

Bramble $\mathcal{B} = \{B_1, B_2, \ldots, B_\ell\}$.

- Naive approach to find hitting sets w. size k (assuming $|\mathcal{B}| > k$):
  - for each $X \subseteq V(D)$ w. $|X| \leq k$,
  - test if X intersects every element of $\mathcal{B}$.



Running time: $\mathcal{O}(n^k) \cdot |\mathcal{B}|$.

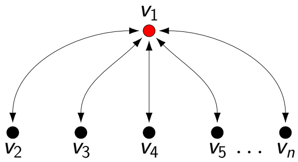
Working with brambles

Naive approach $\mathcal{O}(n^k) \cdot |\mathcal{B}|$.

Working with brambles

Naive approach $\mathcal{O}(n^k) \cdot |\mathcal{B}|$.

- Not ideal: brambles of *small order* can have *exponential size*.

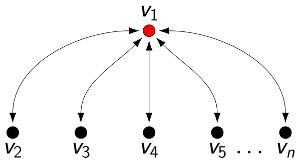


- $\mathcal{B}_{v_1}$ ={all induced subgraphs containing v_1} is a bramble of *order* 1 and *size* 2^{n-1}.

Working with brambles

Naive approach $\mathcal{O}(n^k) \cdot |\mathcal{B}|$.

- Not ideal: brambles of *small order* can have *exponential size*.
- EASY: Haven of order $k \implies$ bramble of order $\lfloor \frac{k}{2} \rfloor + 1$ and size $n^{\mathcal{O}(k)}$.

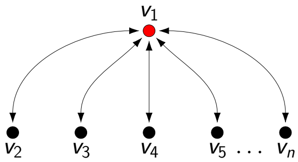


- $\mathcal{B}_{v_1}$ ={all induced subgraphs containing v_1} is a bramble of *order* 1 and *size* 2^{n-1}.

Working with brambles

Naive approach $\mathcal{O}(n^k) \cdot |\mathcal{B}|$.

- Not ideal: brambles of *small order* can have *exponential size*.
- EASY: Haven of order $k \implies$ bramble of order $\lfloor \frac{k}{2} \rfloor + 1$ and size $n^{\mathcal{O}(k)}$.
  - Naive approach $\implies$ XP algorithm to find hitting sets.

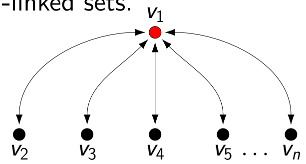


- $\mathcal{B}_{v_1}$ ={all induced subgraphs containing v_1} is a bramble of *order* 1 and *size* 2^{n-1}.

Working with brambles

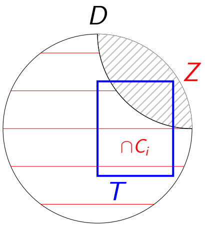
Naive approach $\mathcal{O}(n^k) \cdot |\mathcal{B}|$.

- Not ideal: brambles of *small order* can have *exponential size*.
- EASY: Haven of order $k \implies$ bramble of order $\lfloor \frac{k}{2} \rfloor + 1$ and size $n^{\mathcal{O}(k)}$.
  - Naive approach $\implies$ XP algorithm to find hitting sets.
- Better bramble from k-linked sets.



- $\mathcal{B}_{v_1}$ ={all induced subgraphs containing v_1} is a bramble of *order* 1 and *size* 2^{n-1}.

Brambles from k-linked sets

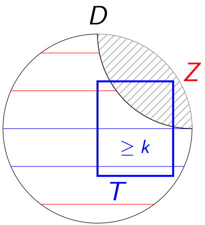


- $|T| \leq 2k-1$.
- T is $(k-1)$-linked $\Rightarrow$ no $k-1$ bal. separator

No $Z \subseteq V(D)$ with $|Z| \leq k-1$ satisfies:

- $|C \cap T| \leq k-1$ for **strong** component C of $D \setminus Z$.

Brambles from k-linked sets

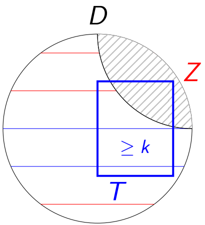


- $|T| \leq 2k - 1$.
- T is $(k-1)$-linked $\Rightarrow$ no $k-1$ bal. separator

$\forall Z \subseteq V(D)$ with $|Z| \leq k - 1$:

- $|C \cap T| \geq k$ for some **strong** component of $D \setminus Z$.

Brambles from k-linked sets



- $|T| \leq 2k - 1$.
- T is $(k-1)$-linked $\Rightarrow$ no $k-1$ bal. separator

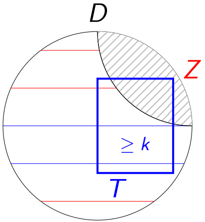
$\forall Z \subseteq V(D)$ with $|Z| \leq k-1$:

- $|C \cap T| \geq k$ for some **strong** component of $D \setminus Z$.

Definition (T-bramble)

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

Brambles from k-linked sets



- $|T| \leq 2k - 1$.
- T is $(k-1)$-linked $\Rightarrow$ no $k-1$ bal. separator

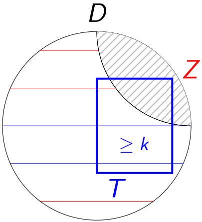
$\forall Z \subseteq V(D)$ with $|Z| \leq k-1$:
- $|C \cap T| \geq k$ for some **strong** component of $D \setminus Z$.

Definition (T-bramble)

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

- $\mathcal{B}_T$ is a bramble: any $B, B' \in \mathcal{B}_T$ intersect since $|T| = 2k-1$.

Brambles from k-linked sets



- $|T| \leq 2k - 1$.
- T is $(k-1)$-linked $\Rightarrow$ no $k-1$ bal. separator

$\forall Z \subseteq V(D)$ with $|Z| \leq k-1$:

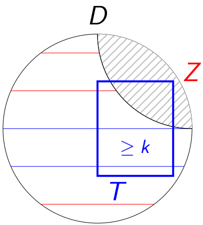
- $|C \cap T| \geq k$ for some **strong** component of $D \setminus Z$.

Definition (T-bramble)

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

- $\mathcal{B}_T$ is a bramble: any $B, B' \in \mathcal{B}_T$ intersect since $|T| = 2k - 1$.
- of order k: any k vertices of T touches every $B \in \mathcal{B}_T$, and T is $k-1$-linked.

Brambles from k-linked sets



- $|T| \leq 2k - 1$.
- T is $(k-1)$-linked $\Rightarrow$ no $k - 1$ bal. separator

$\forall Z \subseteq V(D)$ with $|Z| \leq k - 1$:

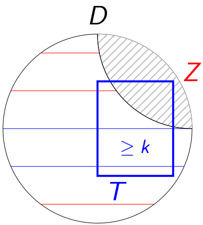
- $|C \cap T| \geq k$ for some **strong** component of $D \setminus Z$.

Definition (T-bramble)

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

- $\mathcal{B}_T$ is a bramble: any $B, B' \in \mathcal{B}_T$ intersect since $|T| = 2k - 1$.
- of order k: any k vertices of T touches every $B \in \mathcal{B}_T$, and T is $k - 1$-linked.
- Skip havens.

Brambles from k-linked sets



- $|T| \leq 2k - 1$.
- T is $(k-1)$-linked $\Rightarrow$ no $k-1$ bal. separator

$\forall Z \subseteq V(D)$ with $|Z| \leq k - 1$:

- $|C \cap T| \geq k$ for some **strong** component of $D \setminus Z$.

Definition (T-bramble)

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

- $\mathcal{B}_T$ is a bramble: any $B, B' \in \mathcal{B}_T$ intersect since $|T| = 2k - 1$.
- of order k: any k vertices of T touches every $B \in \mathcal{B}_T$, and T is $k-1$-linked.
- Skip havens.
- Description: T.

Bramble over $(k-1)$-linket set T

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

Description: T $\Longrightarrow$ Work with $\mathcal{B}_T$ without explicitly looking at its elements.

- Is $D[X] \in \mathcal{B}_T$?
  - Is $D[X]$ **strongly** connected and $|X \cap T| \geq k$? $\Rightarrow$ polynomial-time.

Bramble over $(k-1)$-linket set T

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

Description: T $\Longrightarrow$ Work with $\mathcal{B}_T$ without explicitly looking at its elements.

- Is $D[X] \in \mathcal{B}_T$?
  - Is $D[X]$ **strongly** connected and $|X \cap T| \geq k$? $\Rightarrow$ polynomial-time.
- Is X a *hitting set* of $\mathcal{B}_T$?
  - Is X a T-balanced separator? $\Rightarrow$ polynomial-time.

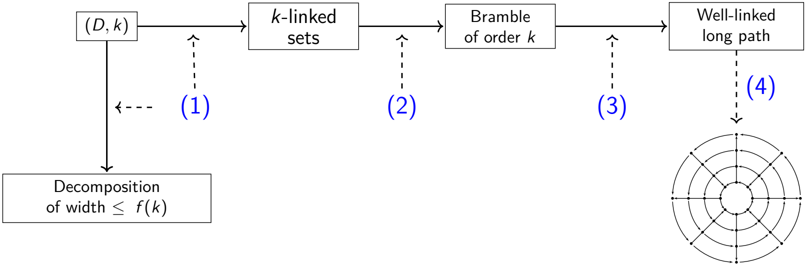
Bramble over $(k-1)$-linket set T

$\mathcal{B}_T = \{B \subseteq D \mid B$ is *induced*, **strongly** connected and $|V(B) \cap T| \geq k\}$.

Description: T $\Longrightarrow$ Work with $\mathcal{B}_T$ without explicitly looking at its elements.

- Is $D[X] \in \mathcal{B}_T$?
  - Is $D[X]$ **strongly** connected and $|X \cap T| \geq k$? $\Rightarrow$ polynomial-time.
- Is X a *hitting set* of $\mathcal{B}_T$?
  - Is X a T-balanced separator? $\Rightarrow$ polynomial-time.
- For $\mathcal{B}' \subseteq \mathcal{B}_T$, is order$(\mathcal{B}') \leq q$?
  - Solvable through T-*partitioning sets* $\Rightarrow$ FPT time (appropriate choices of $\mathcal{B}'$).
    - ⋆ (*Partitioning sets* generalize *balanced separators*).

Directed Grid Theorem: constructive proof (*III*)



(1) Haven vs Decomposition in XP time.

(2) Haven $\Rightarrow$ Bramble of *size* $n^{\mathcal{O}(k)}$.

(3) Bramble $\Rightarrow$ Well-linked long path: working with hitting sets (XP time).

✓(1) *k-linked sets* vs Decomposition in FPT time.

✓(2) *k-linked sets* $\Rightarrow$ Bramble that is *easier* to work with.

(3) Bramble $\Rightarrow$ Well-linked long path: working with hitting sets (FPT time).

Step 1: Hitting set path

Find a path P that hits $\mathcal{B}_T$ in *polynomial-time*.

Step 1: Hitting set path

Find a path P that hits $\mathcal{B}_T$ in *polynomial-time*.

Step 2: Split $\mathcal{B}_T$

Use P to *split* $\mathcal{B}_T$ into "sub-brambles" of adequate size.

Step 1: Hitting set path

Find a path P that hits $\mathcal{B}_T$ in *polynomial-time*.

Step 2: Split $\mathcal{B}_T$

Use P to *split* $\mathcal{B}_T$ into "sub-brambles" of adequate size.

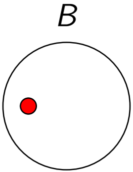
Step 1 + Step 2 $\Longrightarrow$ Well-linked set in P.

Step 1

- GOAL: Find a path P that is hitting set of $\mathcal{B}_T$.

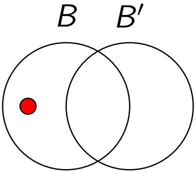
Step 1

- GOAL: Find a path P that is hitting set of $\mathcal{B}_T$.
  - Start with $B \in \mathcal{B}_T$ and $P = v \in B$.



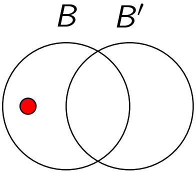
Step 1

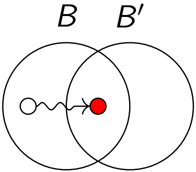
- GOAL: Find a path P that is hitting set of $\mathcal{B}_T$.
  - Start with $B \in \mathcal{B}_T$ and $P = v \in B$.
  - If P does not hit $B' \in \mathcal{B}_T$, improve P (Key ingredient 1).



Step 1

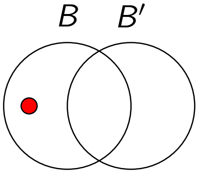
- GOAL: Find a path P that is hitting set of $\mathcal{B}_T$.
  - Start with $B \in \mathcal{B}_T$ and $P = v \in B$.
  - If P does not hit $B' \in \mathcal{B}_T$, improve P (Key ingredient 1).

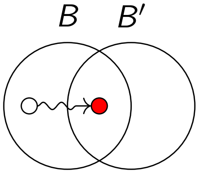




Step 1

- GOAL: Find a path P that is hitting set of $\mathcal{B}_T$.
  - Start with $B \in \mathcal{B}_T$ and $P = v \in B$.
  - If P does not hit $B' \in \mathcal{B}_T$, improve P (Key ingredient 1).
  - Iterate until hitting set.





Key ingredient 1

Is Z a hitting set of $\mathcal{B}_T$?

EASY: Does $D \setminus Z$ contains a **strong** component intersecting k vertices of T?

Step 2 (key ingredient)

Definition (Bramble intersecting X)

$\mathcal{B}_T(X) = \{B \in \mathcal{B}_T \mid B \textit{ intersects } X\}$.

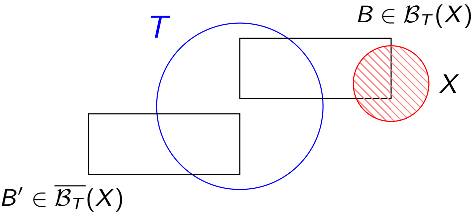
Step 2 (key ingredient)

Definition (Bramble intersecting X)

$\mathcal{B}_T(X) = \{B \in \mathcal{B}_T \mid B \text{ } \textit{intersects} \text{ } X\}$.

Definition (Bramble avoiding X)

$\overline{\mathcal{B}_T}(X) = \{B \in \mathcal{B}_T \mid B \text{ is } \textit{disjoint} \text{ from } X\}$.



Step 2 (key ingredient)

Definition (Bramble intersecting X)

$\mathcal{B}_T(X) = \{B \in \mathcal{B}_T \mid B$ *intersects* $X\}$.

Definition (Bramble avoiding X)

$\overline{\mathcal{B}_T}(X) = \{B \in \mathcal{B}_T \mid B$ is *disjoint* from $X\}$.

Key ingredient

Is $\text{order}(\mathcal{B}_T(X)) \geq k'$?

Not easy. However,

Step 2 (key ingredient)

Definition (Bramble intersecting X)

$\mathcal{B}_T(X) = \{B \in \mathcal{B}_T \mid B$ *intersects* $X\}$.

Definition (Bramble avoiding X)

$\overline{\mathcal{B}_T}(X) = \{B \in \mathcal{B}_T \mid B$ is *disjoint* from $X\}$.

Key ingredient

Is $\text{order}(\mathcal{B}_T(X)) \geq k'$?

Not easy. However,

$$\text{order}(\mathcal{B}_T(X)) + \text{order}(\overline{\mathcal{B}_T}(X)) \geq \text{order}(\mathcal{B}_T)$$

Step 2 (key ingredient)

Definition (Bramble intersecting X)

$\mathcal{B}_T(X) = \{B \in \mathcal{B}_T \mid B$ *intersects* $X\}$.

Definition (Bramble avoiding X)

$\overline{\mathcal{B}_T}(X) = \{B \in \mathcal{B}_T \mid B$ is *disjoint* from $X\}$.

Key ingredient

Is $\text{order}(\mathcal{B}_T(X)) \geq k'$?

Not easy. However,

$$\text{order}(\mathcal{B}_T(X)) + \text{order}(\overline{\mathcal{B}_T}(X)) \geq \text{order}(\mathcal{B}_T)$$

- Estimate $\text{order}(\mathcal{B}_T(X))$ by solving $\text{order}(\overline{\mathcal{B}_T}(X))$.

Step 2 (key ingredient)

Definition (Bramble intersecting X)

$\mathcal{B}_T(X) = \{B \in \mathcal{B}_T \mid B$ *intersects* $X\}$.

Definition (Bramble avoiding X)

$\overline{\mathcal{B}_T}(X) = \{B \in \mathcal{B}_T \mid B$ is *disjoint* from $X\}$.

Key ingredient

Is order$(\mathcal{B}_T(X)) \geq k'$?
▷ FPT time through Partitioning sets.

Not easy. However,

$$\text{order}(\mathcal{B}_T(X)) + \text{order}(\overline{\mathcal{B}_T}(X)) \geq \text{order}(\mathcal{B}_T)$$

- Estimate order$(\mathcal{B}_T(X))$ by solving order$(\overline{\mathcal{B}_T}(X))$.
- Solve order$(\overline{\mathcal{B}_T}(X))$ through *partitioning sets* $\Rightarrow$ FPT time.

Splitting $\mathcal{B}_T$ (First iteration, simplified)

- GOAL: Split $\mathcal{B}_T$ in "sub-brambles" of *adequate order q*, *intersecting* P.
- Path P hitting $\mathcal{B}_T$.
- Find P_1 minimum initial subpath of P s.t. $\text{order}(\mathcal{B}_T(P_1)) \geq q$.

Splitting $\mathcal{B}_T$ (First iteration, simplified)

- GOAL: Split $\mathcal{B}_T$ in "sub-brambles" of *adequate order q*, *intersecting* P.
- Path P hitting $\mathcal{B}_T$.
- Find P_1 minimum initial subpath of P s.t. $\text{order}(\mathcal{B}_T(P_1)) \geq q$.
  - Iteratively: add a vertex, test if $\text{order}(\mathcal{B}_T(P_1)) \geq q$.

Splitting $\mathcal{B}_T$ (First iteration, simplified)

- GOAL: Split $\mathcal{B}_T$ in "sub-brambles" of *adequate order* q, *intersecting* P.
- Path P hitting $\mathcal{B}_T$.
- Find P_1 minimum initial subpath of P s.t. $\text{order}(\mathcal{B}_T(P_1)) \geq q$.
  - Iteratively: add a vertex, test if $\text{order}(\mathcal{B}_T(P_1)) \geq q$.

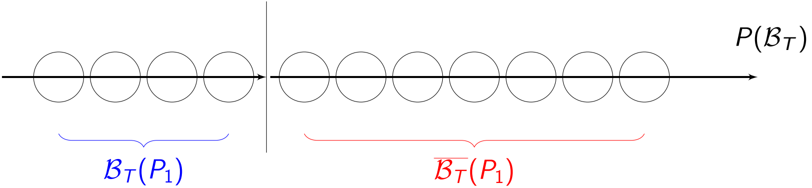
Splitting $\mathcal{B}_T$ (First iteration, simplified)

- GOAL: Split $\mathcal{B}_T$ in "sub-brambles" of *adequate order q*, *intersecting* P.
- Path P hitting $\mathcal{B}_T$.
- Find P_1 minimum initial subpath of P s.t. $\text{order}(\mathcal{B}_T(P_1)) \geq q$.
  - Iteratively: add a vertex, test if $\text{order}(\mathcal{B}_T(P_1)) \geq q$.

Splitting $\mathcal{B}_T$ (First iteration, simplified)

- GOAL: Split $\mathcal{B}_T$ in "sub-brambles" of *adequate order* q, *intersecting* P.
- Path P hitting $\mathcal{B}_T$.
- Find P_1 minimum initial subpath of P s.t. $\text{order}(\mathcal{B}_T(P_1)) \geq q$.
  - Iteratively: add a vertex, test if $\text{order}(\mathcal{B}_T(P_1)) \geq q$.

$$\text{order}(\mathcal{B}_T(P_1)) \geq \text{order}(\mathcal{B}_T) - \text{order}(\mathcal{B}_T(\overline{P_1}))$$



Lemma

In FPT time:

$$\textit{Given } P \textit{ hitting } \mathcal{B}_T, \text{order}(\mathcal{B}_T) \geq (k+1)\left(\left\lfloor \frac{k}{2} \right\rfloor + 1\right) - 1$$

Lemma

In FPT time:

$$\textit{Given } P \textit{ hitting } \mathcal{B}_T,\ \mathrm{order}(\mathcal{B}_T) \geq (k+1)\left(\left\lfloor \frac{k}{2} \right\rfloor + 1\right) - 1$$

⇓

k disjoint "sub-brambles" of order $\geq \left\lfloor \frac{k}{2} \right\rfloor$ intersecting subpaths of P.

Lemma

In FPT time:

$$\text{Given } P \text{ hitting } \mathcal{B}_T, \text{order}(\mathcal{B}_T) \geq (k+1)\left(\left\lfloor \frac{k}{2} \right\rfloor + 1\right) - 1$$

⇓

$$k \text{ disjoint “sub-brambles” of order} \geq \left\lfloor \frac{k}{2} \right\rfloor \text{ intersecting subpaths of } P.$$

- Use sub-brambles to show that P contains *well-linked set* of order k.

Lemma

In FPT time:

$$\textit{Given } P \textit{ hitting } \mathcal{B}_T, \text{order}(\mathcal{B}_T) \geq (k+1)\left(\left\lfloor \frac{k}{2} \right\rfloor + 1\right) - 1$$

k disjoint "sub-brambles" of order $\geq \left\lfloor \frac{k}{2} \right\rfloor$ intersecting subpaths of P.

- Use sub-brambles to show that P contains *well-linked set* of order k.
  - Used to construct highly-connected *system of paths*.

K. Kawarabayashi and S. Kreutzer.
The Directed Grid Theorem
STOC, 2015

- order of $f(k)$ unkown in general, roughly k^6 in planar digraphs.

M. Hatzel, K. Kawarabayashi, and S. Kreutzer.
Polynomial Planar Directed Grid Theorem
SODA'19

- order of $f(k)$ unkown in general, roughly k^6 in planar digraphs.
- Find cylindrical grid or decomposition in FPT time.

- order of $f(k)$ unkown in general, roughly k^6 in planar digraphs.
- Find cylindrical grid or decomposition in FPT time.
- Decomposition is supposed to be a tool.

- order of $f(k)$ unkown in general, roughly k^6 in planar digraphs.
- Find cylindrical grid or decomposition in FPT time.
- Decomposition is supposed to be a tool.
- Most interesting problems are W[1]-hard on graphs of bounded directed tree width.

- order of $f(k)$ unkown in general, roughly k^6 in planar digraphs.
- Find cylindrical grid or decomposition in FPT time.
- Decomposition is supposed to be a tool.
- Most interesting problems are W[1]-hard on graphs of bounded directed tree width.
- Is there an interesting problem can be shown FPT using this?

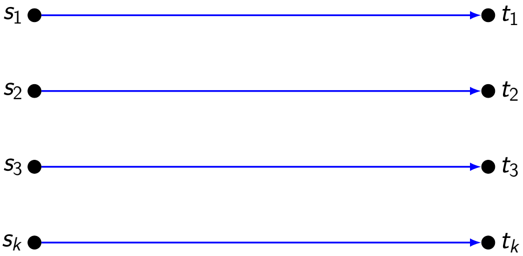
Applications

- H, **strongly** *connected*, has Erdős-Pósa property $\Leftrightarrow$ H is *butterfly* (or topological) minor of a cylindrical grid.

S. Amiri, K. Kawarabayashi, S. Kreutzer and P. Wollan.
The Erdős-Pósa property for directed graphs
CoRR abs/1603.02504, 2016

Applications

- H, **strongly** *connected*, has Erdős-Pósa property $\Leftrightarrow$ H is *butterfly* (or topological) minor of a cylindrical grid.
- Asymmetrical version of DIRECTED DISJOINT PATHS (DDP, k pairs):



Applications

- H, **strongly** *connected*, has Erdős-Pósa property $\Leftrightarrow$ H is *butterfly* (or topological) minor of a cylindrical grid.
- Asymmetrical version of DIRECTED DISJOINT PATHS (DDP, k pairs):
  - *Positive* answer for DDP with congestion c or *negative* answer for DDP.

K. Kawarabayashi and S. Kreutzer.
The Directed Grid Theorem
STOC, 2015

Applications

- H, **strongly** *connected*, has Erdős-Pósa property $\Leftrightarrow$ H is *butterfly* (or topological) minor of a cylindrical grid.
- Asymmetrical version of DIRECTED DISJOINT PATHS (DDP, k pairs):
  - *Positive* answer for DDP with congestion c or *negative* answer for DDP.
  - XP algorithm using Directed Grid Theorem.
  - Claimed to be W[1]-hard, hard to verify.

K. Kawarabayashi and S. Kreutzer.
The Directed Grid Theorem
STOC, 2015

Applications

- H, **strongly** *connected*, has Erdős-Pósa property $\Leftrightarrow$ H is *butterfly* (or topological) minor of a cylindrical grid.
- Asymmetrical version of Directed Disjoint Paths (DDP, k pairs):
  - *Positive* answer for DDP with congestion c or *negative* answer for DDP.
  - XP algorithm using Directed Grid Theorem.
  - Claimed to be W[1]-hard, hard to verify.
- DDP with congestion $= 2$ in $(36k^3 + 2k)$-**strongly** connected digraphs.

Edwards, K., Muzi, I. and Wollan, P.
Half-integral linkages in highly connected directed graphs
In Proc. of the 25th Annual European Symposium on Algorithms (ESA), 2017

Applications

- H, **strongly** *connected*, has Erdős-Pósa property $\Leftrightarrow$ H is *butterfly* (or topological) minor of a cylindrical grid.
- Asymmetrical version of DIRECTED DISJOINT PATHS (DDP, k pairs):
  - *Positive* answer for DDP with congestion c or *negative* answer for DDP.
  - XP algorithm using Directed Grid Theorem.
  - Claimed to be W[1]-hard, hard to verify.
- DDP with congestion $= 2$ in $(36k^3 + 2k)$-**strongly** connected digraphs.
  - Solution in XP time if directed tree-width is bounded.
  - Finds solution in XP time using procedures that we show how to do in FPT time.

Edwards, K., Muzi, I. and Wollan, P.
Half-integral linkages in highly connected directed graphs
In Proc. of the 25th Annual European Symposium on Algorithms (ESA), 2017

Applications

- H, **strongly** *connected*, has Erdős-Pósa property $\Leftrightarrow$ H is *butterfly* (or topological) minor of a cylindrical grid.
- Asymmetrical version of DIRECTED DISJOINT PATHS (DDP, k pairs):
  - *Positive* answer for DDP with congestion c or *negative* answer for DDP.
  - XP algorithm using Directed Grid Theorem.
  - Claimed to be W[1]-hard, hard to verify.
- DDP with congestion $= 2$ in $(36k^3 + 2k)$-**strongly** connected digraphs.
  - Solution in XP time if directed tree-width is bounded.
  - Finds solution in XP time using procedures that we show how to do in FPT time.
- The Directed Flat Wall Theorem recently proved.

A. C. Giannopoulou, K. Kawarabayashi, S. Kreutzer, and O. Kwon.
The Directed Flat Wall Theorem
SODA'20

Applications

- H, **strongly** *connected*, has Erdős-Pósa property ⇔ H is *butterfly* (or topological) minor of a cylindrical grid.
- Asymmetrical version of DIRECTED DISJOINT PATHS (DDP, k pairs):
  - *Positive* answer for DDP with congestion c or *negative* answer for DDP.
  - XP algorithm using Directed Grid Theorem.
  - Claimed to be W[1]-hard, hard to verify.
- DDP with congestion $= 2$ in $(36k^3 + 2k)$-**strongly** connected digraphs.
  - Solution in XP time if directed tree-width is bounded.
  - Finds solution in XP time using procedures that we show how to do in FPT time.
- The Directed Flat Wall Theorem recently proved.
- Courcelle-like meta-theorem w.r.t. directed tree-width (XP time).

M. Oliveira.
An algorithmic metatheorem for directed treewidth
DAM'16

Brambles with constant congestion

Bramble with *congestion* s $\implies$ every vertex in $\leq s$ elements of the bramble.

- Work by Edwards et al. + our results $\implies$ bramble of congestion 2 in FPT time when dtw $\geq f(k)$.

Brambles with constant congestion

Bramble with *congestion s* $\implies$ every vertex in $\leq s$ elements of the bramble.

- Work by Edwards et al. + our results $\implies$ bramble of congestion 2 in FPT time when $\mathrm{dtw} \geq f(k)$.
  - However, $f(k)$ is too large, cylindrical grid $\supseteq$ model of bramble of congestion 2.

Edwards, K., Muzi, I. and Wollan, P.
Half-integral linkages in highly connected directed graphs
In Proc. of the 25th Annual European Symposium on Algorithms (ESA), 2017

Brambles with constant congestion

Bramble with *congestion s* $\implies$ every vertex in $\leq s$ elements of the bramble.

- Work by Edwards et al. + our results $\implies$ bramble of congestion 2 in FPT time when $\text{dtw} \geq f(k)$.
  - However, $f(k)$ is too large, cylindrical grid $\supseteq$ model of bramble of congestion 2.
- Very recent work shows that

  $\text{dtw} \geq t = \mathcal{O}(k^{48} \log^{13} k) \implies$ bramble of congestion ≤ 8.

T. Masařík, M. Pilipczuk, P. Rzążewski, and M. Sorge.
Constant congestion brambles in directed graphs
Manuscript, 2021, available at CoRR abs/2103.08445

Brambles with constant congestion

Bramble with *congestion s* $\implies$ every vertex in $\leq s$ elements of the bramble.

- Work by Edwards et al. + our results $\implies$ bramble of congestion 2 in FPT time when $\text{dtw} \geq f(k)$.
  - However, $f(k)$ is too large, cylindrical grid $\supseteq$ model of bramble of congestion 2.
- Very recent work shows that
  $\text{dtw} \geq t = \mathcal{O}(k^{48} \log^{13} k) \implies$ bramble of congestion ≤ 8.
  - Starts with path-system we can construct in FPT time.

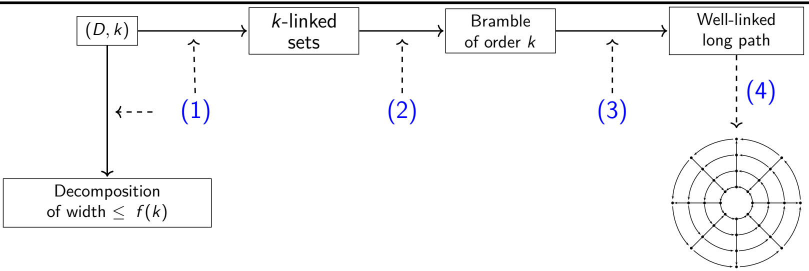
T. Masařík, M. Pilipczuk, P. Rzążewski, and M. Sorge.
Constant congestion brambles in directed graphs
Manuscript, 2021, available at CoRR abs/2103.08445

Brambles with constant congestion

Bramble with *congestion s* $\implies$ every vertex in $\leq s$ elements of the bramble.

- Work by Edwards et al. + our results $\implies$ bramble of congestion 2 in FPT time when $\text{dtw} \geq f(k)$.
  - However, $f(k)$ is too large, cylindrical grid $\supseteq$ model of bramble of congestion 2.
- Very recent work shows that
  $\text{dtw} \geq t = \mathcal{O}(k^{48} \log^{13} k) \implies$ bramble of congestion ≤ 8.
  - Starts with path-system we can construct in FPT time.
  - *Question:* can this bramble be constructed in FPT time?

THANKS!



(1) Haven vs Decomposition in XP time.

(2) Haven $\Rightarrow$ Bramble of *size* $n^{\mathcal{O}(k)}$.

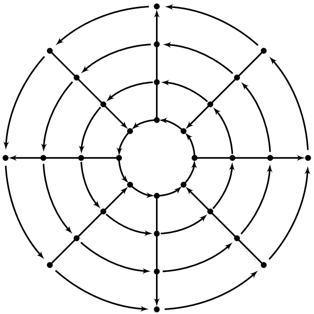
(3) Bramble $\Rightarrow$ Well-linked long path: working with hitting sets (XP time).

✓(1) *k-linked sets* vs Decomposition in FPT time.

✓(2) *k-linked sets* $\Rightarrow$ Bramble that is *easier* to work with.

✓(3) Bramble $\Rightarrow$ Well-linked long path: working with hitting sets (FPT time).

Cylindrical Grid



Cylindrical Grid of order k

- k cycles, same direction.
- $2k$ alternating paths.

Partitioning sets

- Given $T \subseteq V(D)$:

Definition ((T, r)-Partitioning Sets)

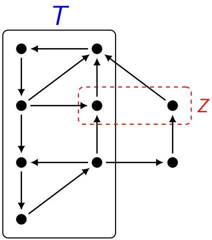
▷ $Z \subseteq V(D)$ is a *(T, r)-partitioning set* if $|T \cap V(C)| \leq r$ for every **strong** component C of $D \setminus Z$.

Partitioning sets

- Given $T \subseteq V(D)$:

Definition ((T, r)-Partitioning Sets)

▷ $Z \subseteq V(D)$ is a *(T, r)-partitioning set* if $|T \cap V(C)| \leq r$ for every **strong** component C of $D \setminus Z$.



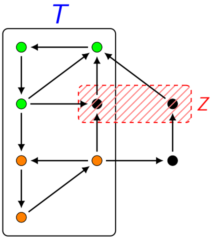
- $|T| = 7, \ r = 3.$

Partitioning sets

- Given $T \subseteq V(D)$:

Definition ((T, r)-Partitioning Sets)

▷ $Z \subseteq V(D)$ is a *(T, r)-partitioning set* if $|T \cap V(C)| \leq r$ for every **strong** component C of $D \setminus Z$.



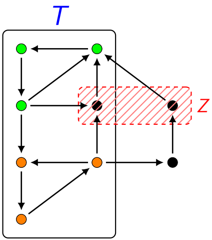
- $|T| = 7$, $r = 3$.
- Z is (T, r)-partitioning set.

Partitioning sets

- Given $T \subseteq V(D)$:

Definition ((T, r)-Partitioning Sets)

▷ $Z \subseteq V(D)$ is a *(T, r)-partitioning set* if $|T \cap V(C)| \leq r$ for every **strong** component C of $D \setminus Z$.



- $|T| = 7$, $r = 3$.
- Z is (T, r)-partitioning set.

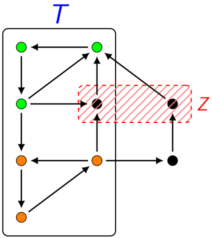
- FPT algorithm when parameterized by $|T|$.

Partitioning sets

- Given $T \subseteq V(D)$:

Definition ((T, r)-Partitioning Sets)

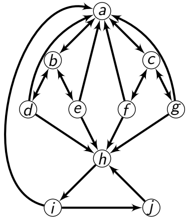
▷ $Z \subseteq V(D)$ is a *(T, r)-partitioning set* if $|T \cap V(C)| \leq r$ for every **strong** component C of $D \setminus Z$.



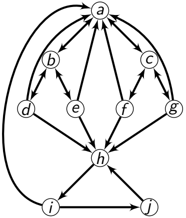
- $|T| = 7,\ r = 3$.
- Z is (T, r)-partitioning set.

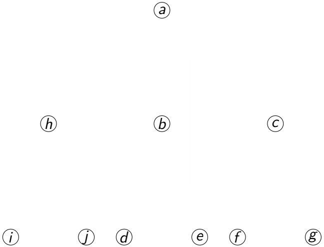
- FPT algorithm when parameterized by $|T|$.
- Question: FPT algorithm when parameterized by $|Z|$?
  - $T = V(D), r = 0 \implies$ FEEDBACK VERTEX SET.

Directed tree-width-I



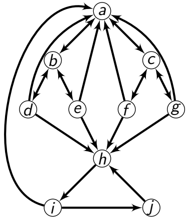
Directed tree-width-I

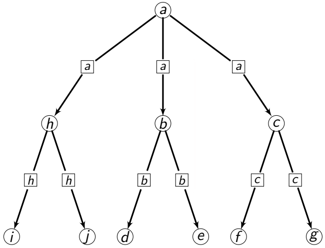




- Place vertices into "bags".

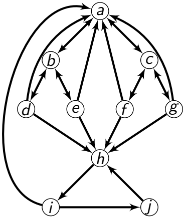
Directed tree-width-I

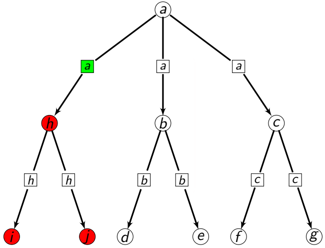




- Place vertices into "bags".
- Set edges as "guards".

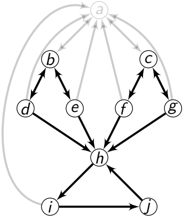
Directed tree-width-I

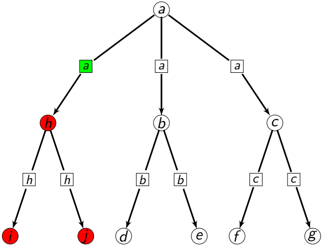




- Place vertices into “bags”.
- Set edges as “guards”.
- $\{a\}$ guards $\{h, i, j\}$.

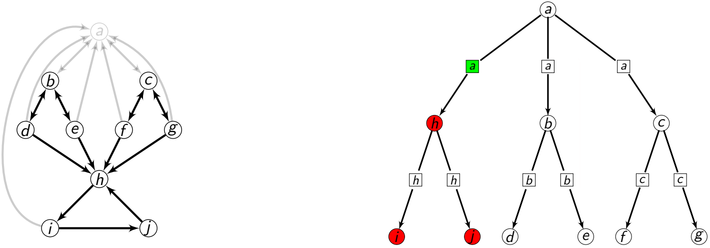
Directed tree-width-I





- Place vertices into "bags".
- Set edges as "guards".
- $\{a\}$ guards $\{h, i, j\}$.

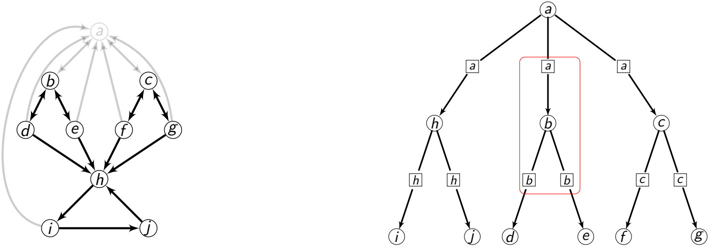
Directed tree-width-I



- Place vertices into "bags".
- Set edges as "guards".
- $\{a\}$ guards $\{h, i, j\}$.

$\Longrightarrow$ No path starting and ending in $\{h, i, j\}$ using vertices from $\{b, c, d, e, f, g\}$ in $G \setminus \{a\}$.

Directed tree-width-I



- Place vertices into "bags".
- Set edges as "guards".
- $\{a\}$ guards $\{h, i, j\}$.

$\Longrightarrow$ No path starting and ending in $\{h, i, j\}$ using vertices from $\{b, c, d, e, f, g\}$ in $G \setminus \{a\}$.

- width = size of largest set of "bag" + adjacent "guards".

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

- R is an arborescence;

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

- R is an arborescence;
- $\mathcal{X} = \{X_e : e \in E(R)\}$;

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

- R is an arborescence;
- $\mathcal{X} = \{X_e : e \in E(R)\}$;
- $\mathcal{W} = \{W_r : r \in V(R)\}$;

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

- R is an arborescence;
- $\mathcal{X} = \{X_e : e \in E(R)\}$;
- $\mathcal{W} = \{W_r : r \in V(R)\}$;
- $\mathcal{X}$ and $\mathcal{W}$ are families of subsets of $V(G)$,

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

- R is an arborescence;
- $\mathcal{X} = \{X_e : e \in E(R)\}$;
- $\mathcal{W} = \{W_r : r \in V(R)\}$;
- $\mathcal{X}$ and $\mathcal{W}$ are families of subsets of $V(G)$,

and

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

- R is an arborescence;
- $\mathcal{X} = \{X_e : e \in E(R)\}$;
- $\mathcal{W} = \{W_r : r \in V(R)\}$;
- $\mathcal{X}$ and $\mathcal{W}$ are families of subsets of $V(G)$,

and

- $\mathcal{W}$ partitions $V(G)$ into non-empty sets;

Directed tree-width-II

- Estimation to how close a directed graph is to a DAG.

Definition (Arboreal decomposition)

An *arboreal decomposition* of a directed graph $G = (V, E)$ is a triple $\mathcal{T} = (R, \mathcal{X}, \mathcal{W})$ where:

- R is an arborescence;
- $\mathcal{X} = \{X_e : e \in E(R)\}$;
- $\mathcal{W} = \{W_r : r \in V(R)\}$;
- $\mathcal{X}$ and $\mathcal{W}$ are families of subsets of $V(G)$,

and

- $\mathcal{W}$ partitions $V(G)$ into non-empty sets;
- if $e \in E(R)$ then $\bigcup\{W_r : r \in V(R), r > e\}$ is guarded by X_e.

Havens

Definition (Haven of order k)

A function $h(Z)$ s.t., for $Z \subseteq V(D)$ with $|Z| \leq k - 1$,

Havens

Definition (Haven of order k)

A function $h(Z)$ s.t., for $Z \subseteq V(D)$ with $|Z| \leq k-1$,

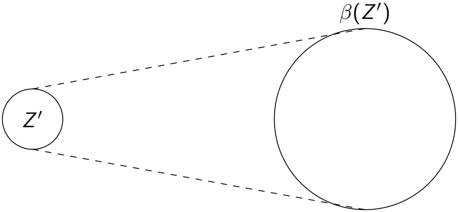
- $h(Z)$ is strongly connected component of $D \setminus Z$; and

Havens

Definition (Haven of order k)

A function $h(Z)$ s.t., for $Z \subseteq V(D)$ with $|Z| \leq k-1$,

- $h(Z)$ is strongly connected component of $D \setminus Z$; and
- $Z' \subseteq Z \implies h(Z) \subseteq h(Z')$.

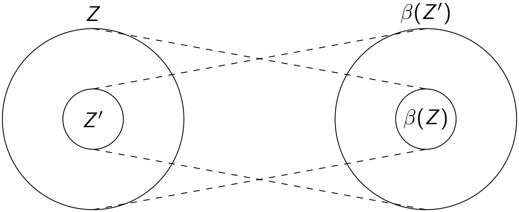


Havens

Definition (Haven of order k)

A function $h(Z)$ s.t., for $Z \subseteq V(D)$ with $|Z| \leq k-1$,

- $h(Z)$ is strongly connected component of $D \setminus Z$; and
- $Z' \subseteq Z \implies h(Z) \subseteq h(Z')$.

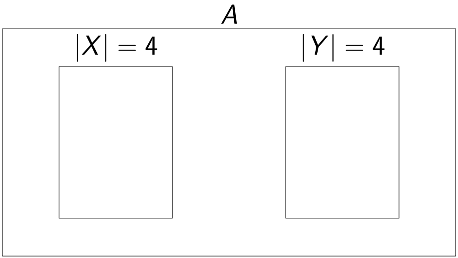


Well-linked sets

Definition (Well-linked sets)

Set $A \subseteq V(D)$ is *well-linked* if:

- For all disjoint $X, Y \subseteq A$ with $|X| = |Y|$,

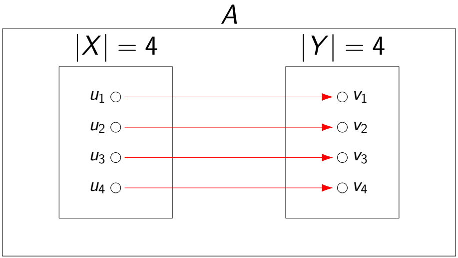


Well-linked sets

Definition (Well-linked sets)

Set $A \subseteq V(D)$ is *well-linked* if:

- For all disjoint $X, Y \subseteq A$ with $|X| = |Y|$,
  - There are $|X|$ pairwise disjoint paths from X to Y.

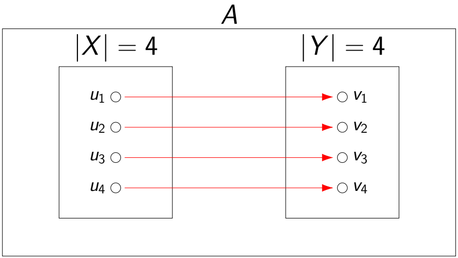


Well-linked sets

Definition (Well-linked sets)

Set $A \subseteq V(D)$ is *well-linked* if:

- For all disjoint $X, Y \subseteq A$ with $|X| = |Y|$,
  - There are $|X|$ pairwise disjoint paths from X to Y.



Well-linked sets

Definition (Well-linked sets)

Set $A \subseteq V(D)$ is *well-linked* if:

- For all disjoint $X, Y \subseteq A$ with $|X| = |Y|$,
  - There are $|X|$ pairwise disjoint paths from X to Y.
- $\text{order}(A) = |A|$.