Hitting and Harvesting Pumpkins

Gwenaël Joret Christophe Paul Ignasi Sau*
Saket Saurabh Stéphan Thomassé

*CNRS, LIRMM, Montpellier, France.

ESA 2011. *Saarbrücken, Germany.*

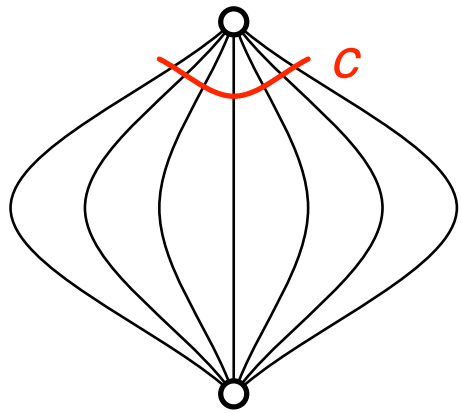
Outline of the talk

Next section is...

Pumpkins

Pumpkins

c-pumpkin:

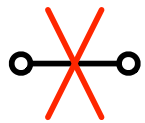


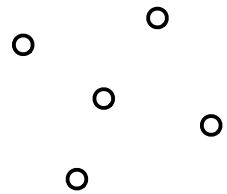
N.B: “graph” = multigraph

Graphs with no c-pumpkin minor

Graphs with no c-pumpkin minor

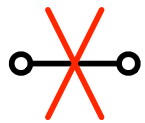
- $c = 1$: empty graphs

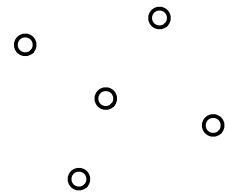




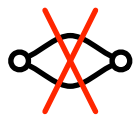
Graphs with no c-pumpkin minor

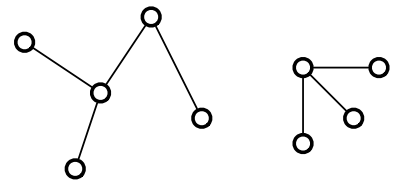
- $c = 1$: empty graphs





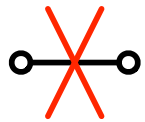
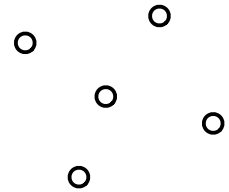
- $c = 2$: forests



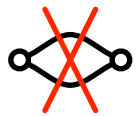
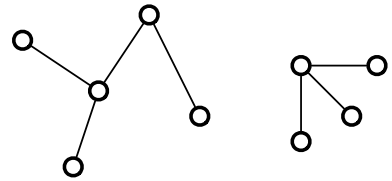


Graphs with no c-pumpkin minor

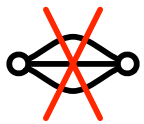
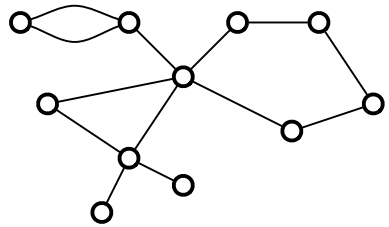
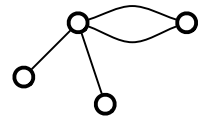
- $c = 1$: empty graphs

- $c = 2$: forests

- $c = 3$: no two cycles share an edge

- etc.

Next subsection is...

Hitting pumpkins

For the whole talk:

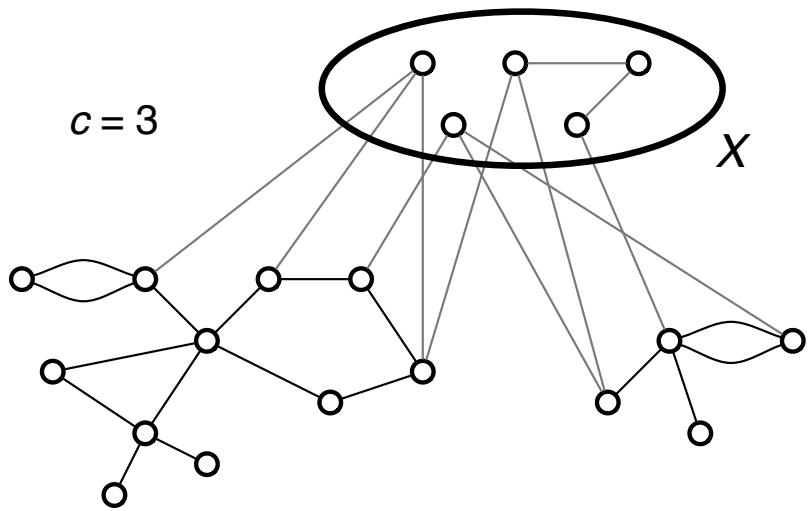
- $c \geqslant 1$ fixed integer
- G input graph
- $n := |V(G)|$

Hitting pumpkins

For the whole talk:

- $c \geqslant 1$ fixed integer
- G input graph
- $n := |V(G)|$

c-pumpkin hitting set:
vertex subset $X \subseteq V(G)$ s.t. $G - X$ has no c-pumpkin minor

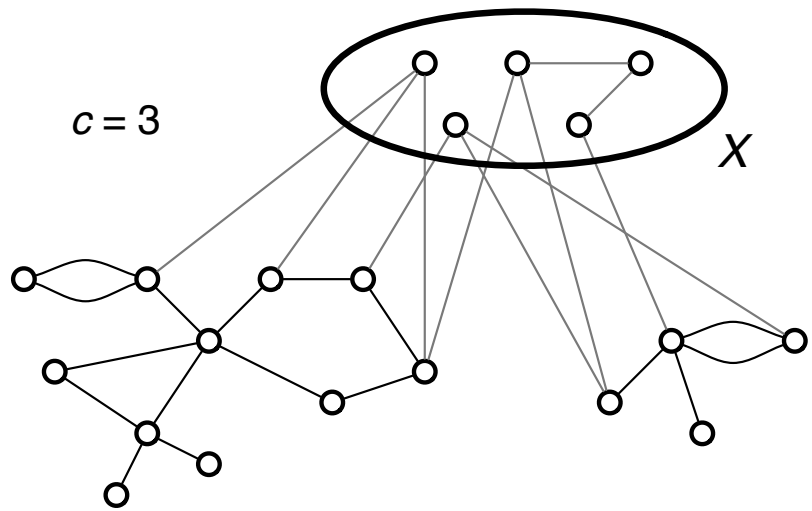


Hitting pumpkins

For the whole talk:

- $c \geqslant 1$ fixed integer
- G input graph
- $n := |V(G)|$

c-pumpkin hitting set:
vertex subset $X \subseteq V(G)$ s.t. $G - X$ has no c-pumpkin minor



Hitting set (or *transversal*) number:
min. size of a c-pumpkin hitting set

c-Pumpkin Hitting Set problem:

Find a c-pumpkin hitting set of minimum size.

NP-hard $\forall c \geqslant 1$

c-Pumpkin Hitting Set problem:
Find a c-pumpkin hitting set of minimum size.

NP-hard $\forall c \geqslant 1$

Interested in **FPT** and **approximation algorithms**

In parameterized version:

- extra parameter k
- goal: decide if $\exists$ c-pumpkin hitting set of size $\leqslant k$

Some special cases

$c = 1$: Vertex Cover

- 2-approximation algorithm
- $O(1.2738^k + kn)$-time FPT algorithm [Chen, Kanj, Xia '10]

Some special cases

$c = 1$: Vertex Cover

- 2-approximation algorithm
- $O(1.2738^k + kn)$-time FPT algorithm [Chen, Kanj, Xia '10]

$c = 2$: Feedback Vertex Set

- 2-approximation algorithms [Bafman, Berman, Fujito '99, Becker & Geiger '96]
- $O(3.83^k \cdot k \cdot n^2)$-time FPT algorithm [Cao, Chen, Liu '10]

Some special cases

$c = 1$: Vertex Cover

- 2-approximation algorithm
- $O(1.2738^k + kn)$-time FPT algorithm [Chen, Kanj, Xia '10]

$c = 2$: Feedback Vertex Set

- 2-approximation algorithms [Bafman, Berman, Fujito '99, Becker & Geiger '96]
- $O(3.83^k \cdot k \cdot n^2)$-time FPT algorithm [Cao, Chen, Liu '10]

$c = 3$: Diamond Hitting Set

- 9-approximation algorithm [Fiorini, Joret, Pietropaoli '10]

$\forall c \geqslant 1$: [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

- $O(\log^{3/2} OPT)$-approximation algorithm
- $2^{O(k \log k)} n^{O(1)}$-time FPT algorithm

$\forall c \geqslant 1$: [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

- $O(\log^{3/2} OPT)$-approximation algorithm
- $2^{O(k \log k)} n^{O(1)}$-time FPT algorithm

Question: Does there exist a $2^{O(k)} n^{O(1)}$-time FPT algorithm $\forall c \geqslant 1$? (called **single-exponential** algorithm)

$\forall c \geqslant 1$: [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

- $O(\log^{3/2} OPT)$-approximation algorithm
- $2^{O(k \log k)} n^{O(1)}$-time FPT algorithm

Question: Does there exist a $2^{O(k)} n^{O(1)}$-time FPT algorithm $\forall c \geqslant 1$? (called **single-exponential** algorithm)

Theorem (Joret, Paul, S., Saurabh, Thomassé)

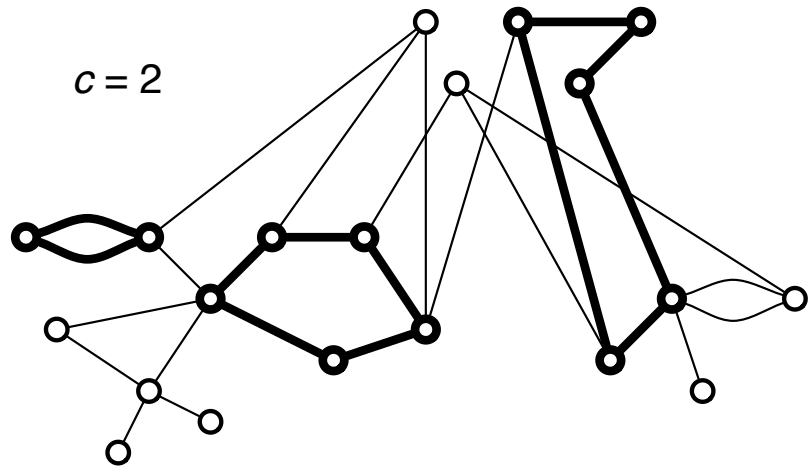
There is a single-exponential FPT algorithm $\forall c \geqslant 1$

N.B: no $2^{o(k)} n^{O(1)}$-time FPT algorithm unless ETH fails

Next subsection is...

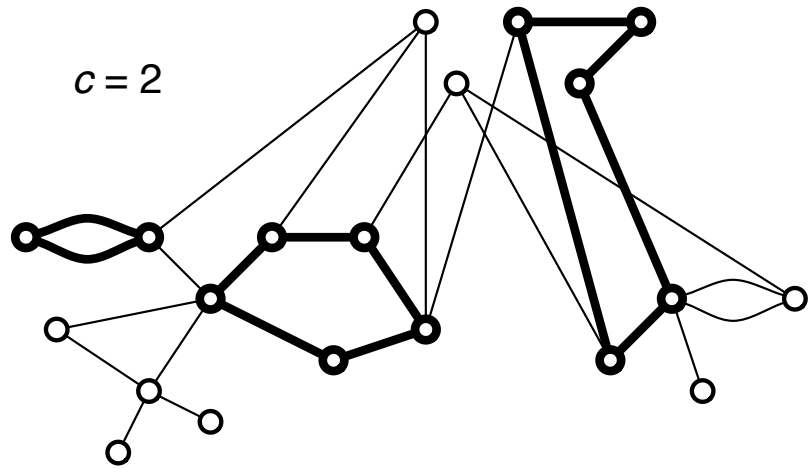
Packing pumpkins

c-pumpkin packing:
collection of vertex-disjoint subgraphs of G, each containing a c-pumpkin minor



Packing pumpkins

c-pumpkin packing:
collection of vertex-disjoint subgraphs of G, each containing a c-pumpkin minor



Packing number: max. cardinality of a c-pumpkin packing

c-Pumpkin Packing problem:

Find a c-pumpkin packing of max. cardinality.

NP-hard $\forall c \geqslant 2$

c-Pumpkin Packing problem:
Find a c-pumpkin packing of max. cardinality.

NP-hard $\forall c \geqslant 2$

$c = 1$: Maximum Matching

c-Pumpkin Packing problem:
Find a c-pumpkin packing of max. cardinality.

NP-hard $\forall c \geqslant 2$

$c = 1$: Maximum Matching

$c = 2$: Maximum Cycle Packing

- $O(\log n)$-approximation algorithm [Krivelevich, Nutov, Salavatipour '07]
- $\Omega(\log^{1/2-\varepsilon} n)$-inapproximability [Friggstad, Salavatipour '11]

c-Pumpkin Packing problem:
Find a c-pumpkin packing of max. cardinality.

NP-hard $\forall c \geqslant 2$

$c = 1$: Maximum Matching

$c = 2$: Maximum Cycle Packing

- $O(\log n)$-approximation algorithm [Krivelevich, Nutov, Salavatipour '07]
- $\Omega(\log^{1/2-\varepsilon} n)$-inapproximability [Friggstad, Salavatipour '11]

Question: approximation algorithms for $c \geqslant 3$?

Approximate min-max relation for packings and hitting sets:

Theorem (Joret, Paul, S., Saurabh, Thomassé)

There exist

- *a c-pumpkin packing $\mathcal{M}$, and*
- *a c-pumpkin hitting set X*

s.t. $|X| \leqslant O_c(\log n) \cdot |\mathcal{M}|$, *for every fixed* $c \geqslant 1$

Approximate min-max relation for packings and hitting sets:

Theorem (Joret, Paul, S., Saurabh, Thomassé)

There exist

- *a c-pumpkin packing $\mathcal{M}$, and*
- *a c-pumpkin hitting set X*

s.t. $|X| \leqslant O_c(\log n) \cdot |\mathcal{M}|$, *for every fixed* $c \geqslant 1$

$\Rightarrow$ $O_c(\log n)$-approximation algorithm for c-Pumpkin Hitting Set and c-Pumpkin Packing for every fixed $c \geqslant 1$

Next section is...

FPT algorithm

Goal: Single-exponential FPT algo for c-Pumpkin Hitting Set

We use the technique of iterative compression

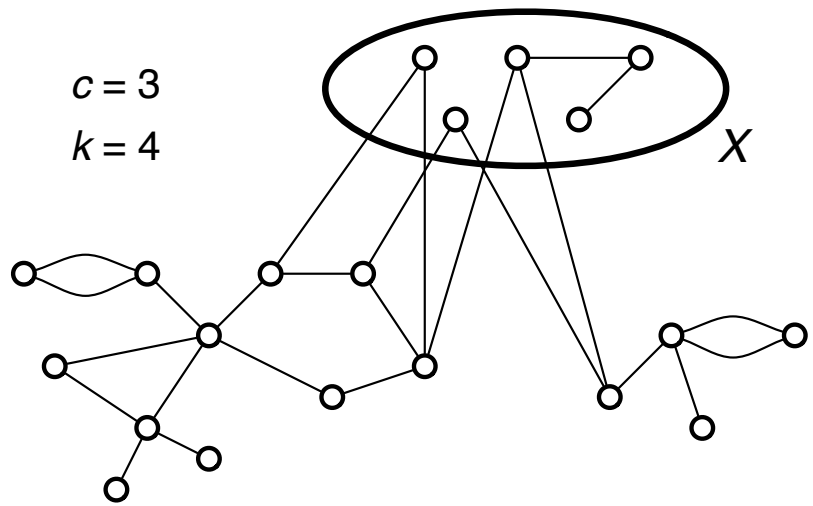
FPT algorithm

Goal: Single-exponential FPT algo for c-Pumpkin Hitting Set

We use the technique of iterative compression

Switch to Disjoint c-Pumpkin Hitting Set problem:

- Input: G, hitting set X with $|X| \leqslant k+1$
- Question: is there a hitting set X' with $|X'| \leqslant k$ that is **disjoint from** X?



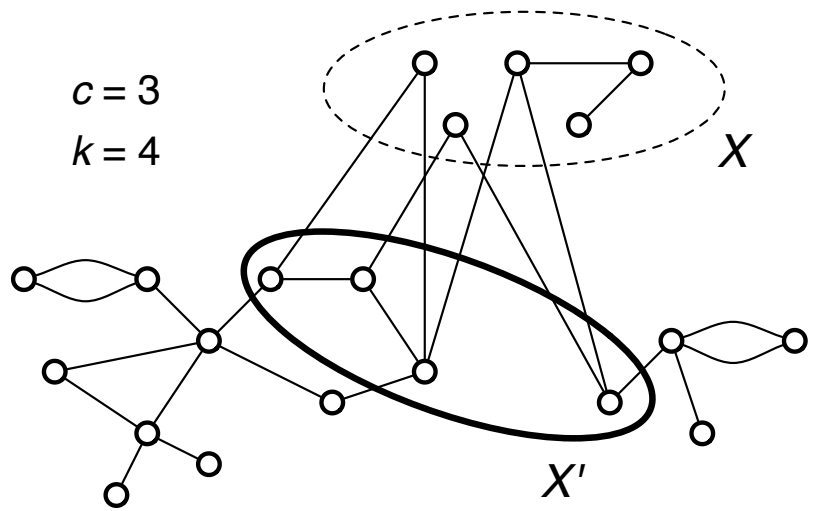
FPT algorithm

Goal: Single-exponential FPT algo for c-Pumpkin Hitting Set

We use the technique of iterative compression

Switch to Disjoint c-Pumpkin Hitting Set problem:

- Input: G, hitting set X with $|X| \leqslant k+1$
- Question: is there a hitting set X' with $|X'| \leqslant k$ that is **disjoint from** X?



Lemma

$d^k \cdot n^{O(1)}$ algorithm for Disjoint c-Pumpkin Hitting Set $\Rightarrow$ *$(d+1)^k \cdot n^{O(1)}$ algorithm for* c-Pumpkin Hitting Set

Main ingredients of FPT algorithm:

- poly-time (simple) ad-hoc reduction rules
- protrusion-based reduction rule
- branching rule (with single-exponential # of subproblems)
- linear kernel in a special case

Next section is...

Approximation algorithms

Goal: Finding

- a c-pumpkin packing $\mathcal{M}$ and
- a c-pumpkin hitting set X

s.t. $|X| \leqslant O_c(\log n) \cdot |\mathcal{M}|$

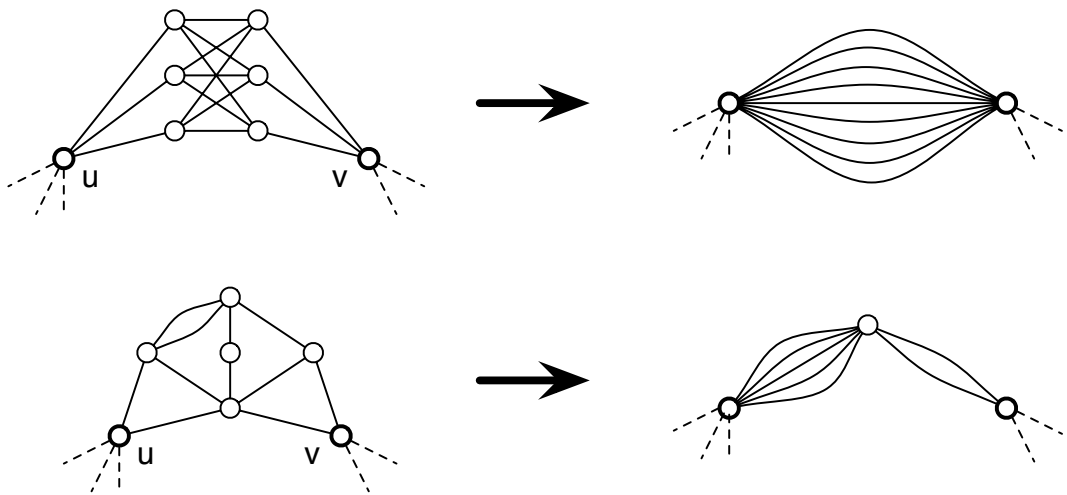
Reduction rules

Goal: smaller graph with same packing and hitting set numbers

Reduction rules

Goal: smaller graph with same packing and hitting set numbers

- u, v minimal 2-separator, C a connected component of $G \setminus \{u, v\}$
- d-pumpkin-model: two disjoint sets of vertices A, B, each inducing a connected subgraph of G, with at least d edges between them
- $\alpha(C, u, v)$: largest integer d such that $G[C, u, v] \setminus uv$ has a d-pumpkin-model $\{A, B\}$ with $u \in A$ and $v \in B$
- $\beta(C, u, v)$: largest integer d such that $G[C, u, v] + uv$ has a d-pumpkin-model $\{A, B\}$ with $u, v \in A$
- Two cases: $\alpha \geqslant \beta$ and $\alpha < \beta$



Small pumpkins

Subgraph small if of size $\leqslant h(c) \cdot \log n$

(where is h some fixed, computable function)

Small pumpkins

Subgraph small if of size $\leqslant h(c) \cdot \log n$
(where is h some fixed, computable function)

$d^*(G) :=$ average degree of underlying simple graph of G

If $d^*(G) \geqslant 2^t$ then $\exists G' \subseteq G$ with $|V(G')| = O_t(\log n)$ s.t. G' contains a K_t-minor [Fiorini, Joret, Theis, Wood '10]

Small pumpkins

Subgraph small if of size $\leqslant h(c) \cdot \log n$
(where is h some fixed, computable function)

$d^*(G) :=$ average degree of underlying simple graph of G

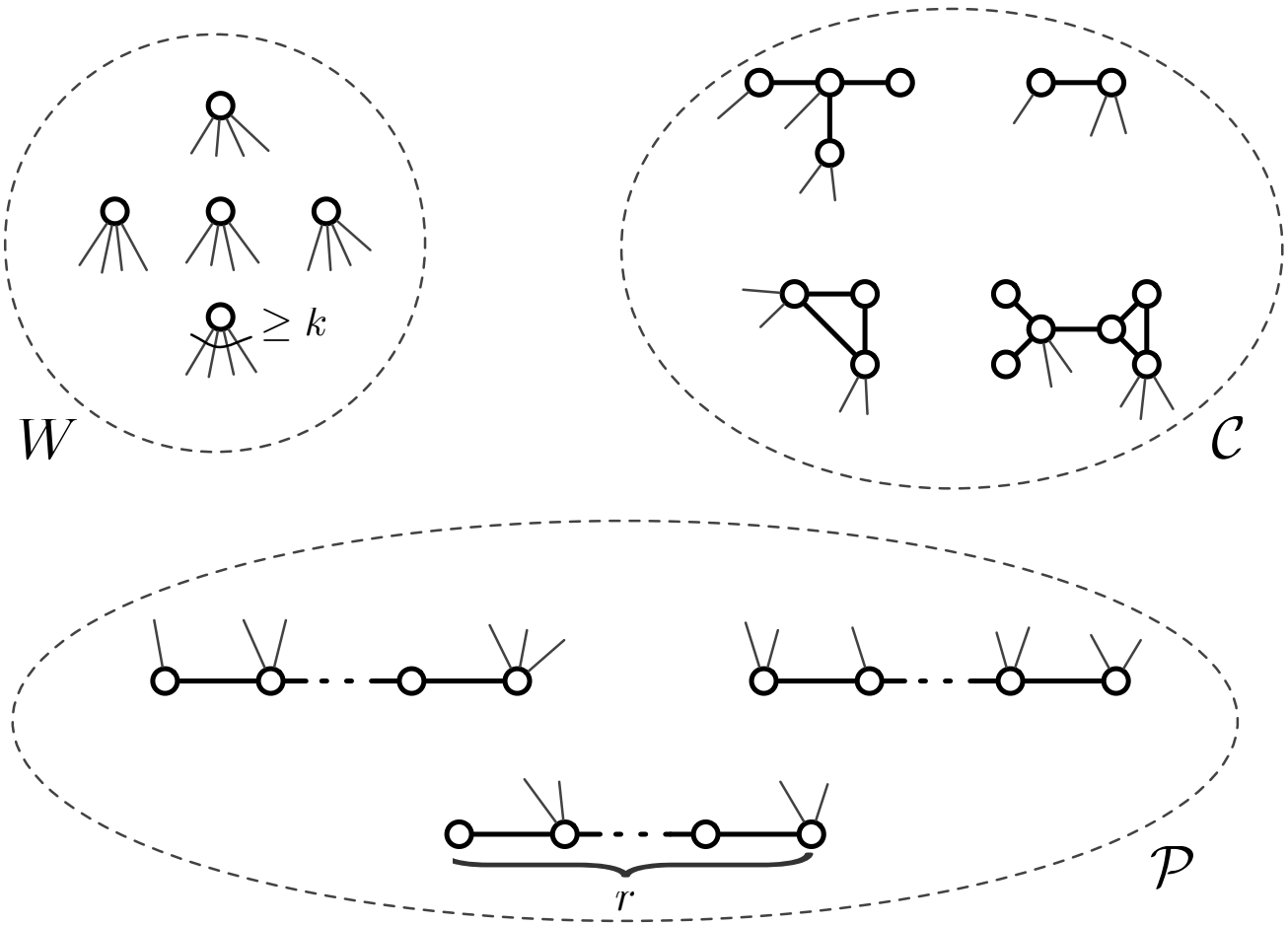
If $d^*(G) \geqslant 2^t$ then $\exists G' \subseteq G$ with $|V(G')| = O_t(\log n)$ s.t. G' contains a K_t-minor [Fiorini, Joret, Theis, Wood '10]

$\Rightarrow$ if $d^*(G) \geqslant 2^{2\sqrt{c}+1}$ then G has a **small** subgraph containing a c-pumpkin minor

Theorem

Either G has a small c-pumpkin minor or some reduction rule can be applied

Proof idea:



Approximation algorithm:

$\mathcal{M} \leftarrow \emptyset$; $X \leftarrow \emptyset$

If G not reduced:

Apply reduction rule on G

Call algorithm on resulting graph

Else:

Compute a small c-pumpkin minor M

Call algorithm on $G \setminus V(M)$, giving a packing $\mathcal{M}'$ and a hitting set X'

$\mathcal{M} \leftarrow \mathcal{M}' \cup \{M\}$

$X \leftarrow X' \cup V(M)$

Next section is...

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds
  - faster algorithms for sparse graphs?

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds
  - faster algorithms for sparse graphs?
  - **Challenging**: deleting at most k vertices from a given graph so that the resulting graph has tree-width bounded by some constant c

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds
  - faster algorithms for sparse graphs?
  - **Challenging**: deleting at most k vertices from a given graph so that the resulting graph has tree-width bounded by some constant c (OK, single-exponential for $c = 2$) [Kim, Paul, Philip '11]

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds
  - faster algorithms for sparse graphs?
  - **Challenging**: deleting at most k vertices from a given graph so that the resulting graph has tree-width bounded by some constant c (OK, single-exponential for $c = 2$) [Kim, Paul, Philip '11]
- ★ we provided an $O_c(\log n)$-approximation algorithm for c-Pumpkin Hitting Set and c-Pumpkin Packing

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds
  - faster algorithms for sparse graphs?
  - **Challenging**: deleting at most k vertices from a given graph so that the resulting graph has tree-width bounded by some constant c (OK, single-exponential for $c = 2$) [Kim, Paul, Philip '11]
- ★ we provided an $O_c(\log n)$-approximation algorithm for c-Pumpkin Hitting Set and c-Pumpkin Packing
  - constant-factor approximation for the hitting version? (so far, such an algorithm is only known for $c \leqslant 3$)

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds
  - faster algorithms for sparse graphs?
  - **Challenging**: deleting at most k vertices from a given graph so that the resulting graph has tree-width bounded by some constant c (OK, single-exponential for $c = 2$) [Kim, Paul, Philip '11]
- ★ we provided an $O_c(\log n)$-approximation algorithm for c-Pumpkin Hitting Set and c-Pumpkin Packing
  - constant-factor approximation for the hitting version? (so far, such an algorithm is only known for $c \leqslant 3$)
  - packing edge-disjoint c-pumpkin models

Conclusions and further research

- ★ we provided an FPT algorithm running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time for c-Pumpkin Hitting Set
  - can we avoid protrusions?
  - optimizing the constants
  - provide lower bounds
  - faster algorithms for sparse graphs?
  - **Challenging**: deleting at most k vertices from a given graph so that the resulting graph has tree-width bounded by some constant c (OK, single-exponential for $c = 2$) [Kim, Paul, Philip '11]
- ★ we provided an $O_c(\log n)$-approximation algorithm for c-Pumpkin Hitting Set and c-Pumpkin Packing
  - constant-factor approximation for the hitting version? (so far, such an algorithm is only known for $c \leqslant 3$)
  - packing edge-disjoint c-pumpkin models
  - find explicit (small?) function for the Erdős-Pósa property

Gràcies!