Kernelization dichotomies for hitting minors under structural parameterizations

Marin Bougeret[1], Eric Brandwein[2] and Ignasi Sau[1]

STACS 2026

[1] LIRMM, University of Montpellier, France
[2] Universidad de Buenos Aires, Argentina

Outline

Structural parameterization

VC|λ (Vertex Cover)

- input :(G, k)
- question :decide if $VC(G) \leq k$ (if there exists a set of k vertices intersecting all edges)
- parameter :$\lambda(G, k) = ..$

Structural parameterization

VC|λ (Vertex Cover)

- input :(G, k)
- question :decide if $VC(G) \leq k$ (if there exists a set of k vertices intersecting all edges)
- parameter :$\lambda(G, k) = ..$

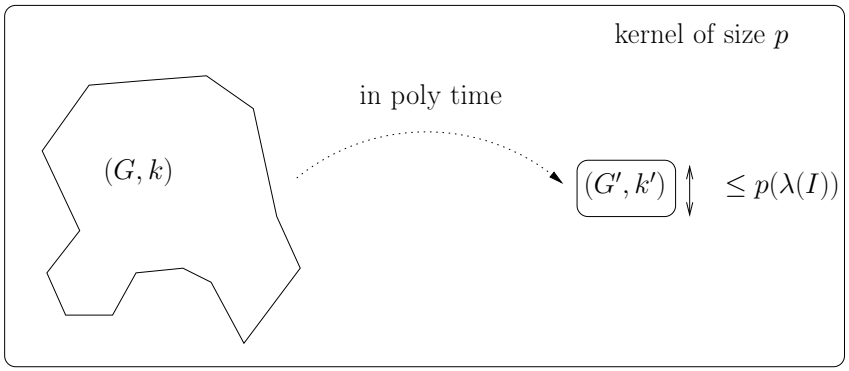
- for FPT algorithm, positive results ($f(\lambda(G, k))n^{\mathcal{O}(1)}$) exists for many structural parameters ($\lambda(G, k) = tw(G)$)

Structural parameterization

VC|λ (Vertex Cover)

- input :(G, k)
- question :decide if $VC(G) \leq k$ (if there exists a set of k vertices intersecting all edges)
- parameter :$\lambda(G, k) = ..$

- for polynomial kernels
  - such parameters are not possible (VC/tw do not admit a poly kernel)
  - but using $\lambda(G, k)$ as distance to triviality measures allow for kernels

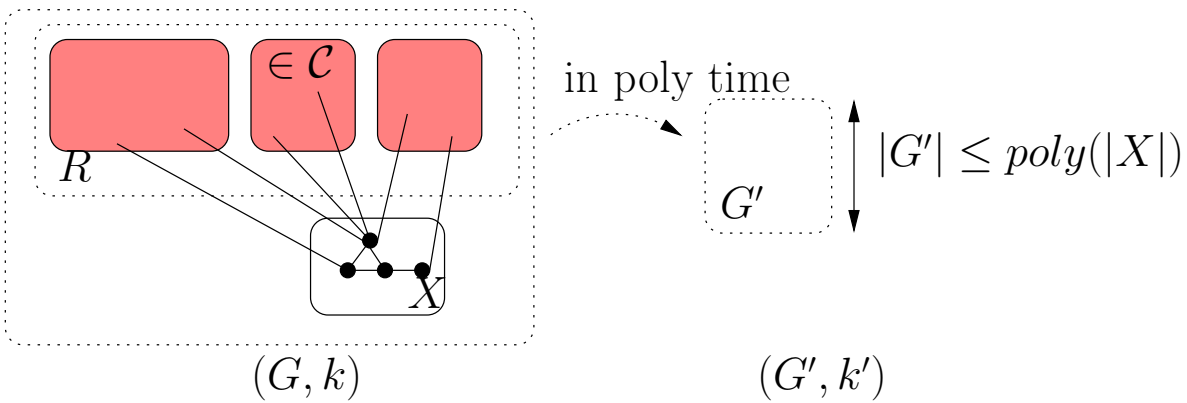


Kernelization parameterized by distance to triviality

Take your favorite trivial class $\mathcal{C}$ ($\mathcal{C}$= independent set, forest):

VC|dist-to-$\mathcal{C}$

- input: (G, X, k) where $R = G \setminus X \in \mathcal{C}$
- question: decide if $VC(G) \leq k$
- parameter: $|X|$

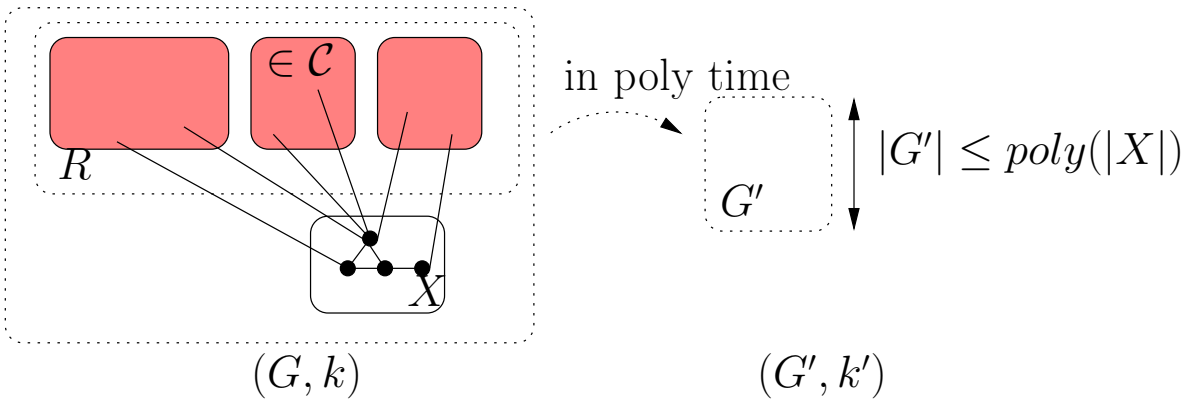


Kernelization parameterized by distance to triviality

Take your favorite trivial class $\mathcal{C}$ ($\mathcal{C}$= independent set, forest):

VC|dist-to-$\mathcal{C}$

- input: (G, X, k) where $R = G \setminus X \in \mathcal{C}$
- question: decide if $VC(G) \leq k$
- parameter: $|X|$



Motivation

$\mathcal{O}(|X|^{\mathcal{O}(1)})$ kernel for VC|dist-to-$\mathcal{C}$ can lead to smaller reduced graph than $\mathcal{O}(k)$ kernel for $VC|k$.

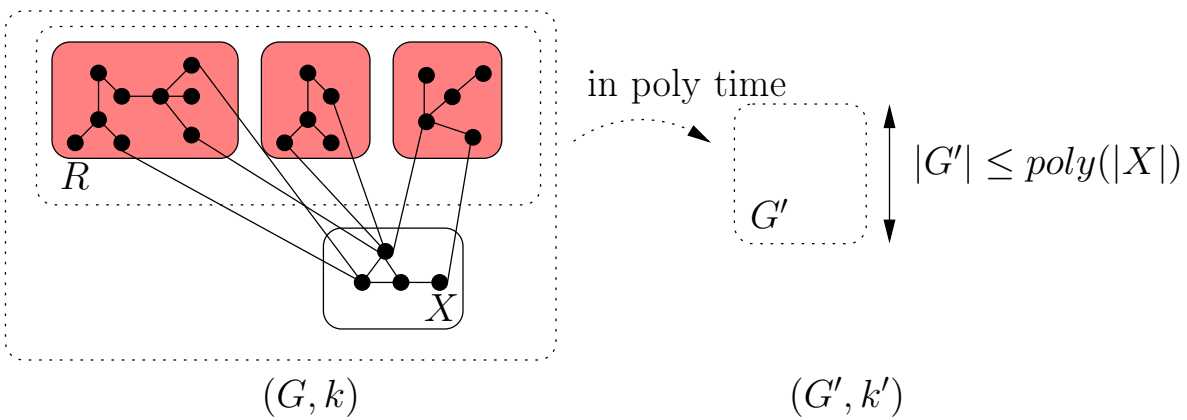
Kernelization parameterized by distance to triviality

First result of that kind for VC from (Bodlaender, Jansen)

VC|dist-to-Forest has a polynomial kernel.

VC|dist-to-Forest

- input: (G, X, k) where $R = G \setminus X$ is a forest
- question: decide if $VC(G) \leq k$
- parameter: $\lambda(G, X, k) = |X|$



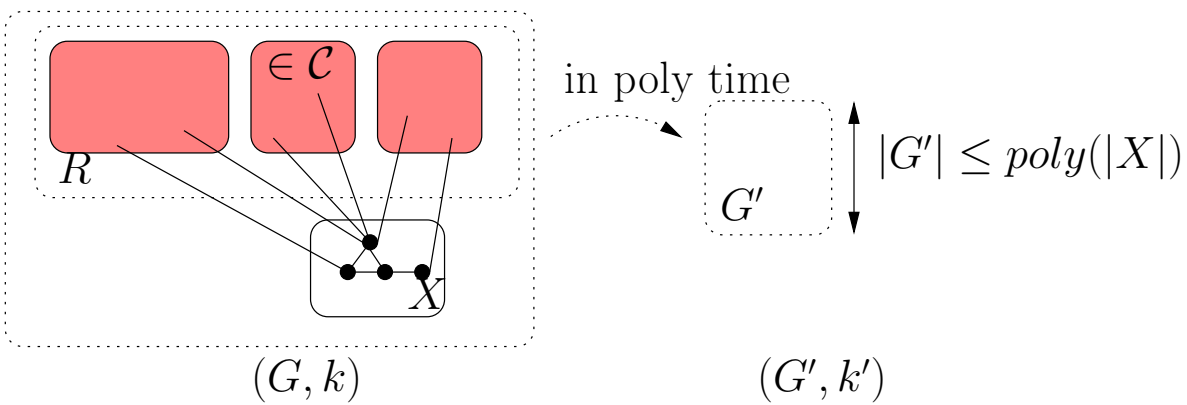
Landscape

New race!

Goal: obtain polynomial kernel VC|dist-to-$\mathcal{C}$ for $\mathcal{C}$ as general as possible.

VC|dist-to-$\mathcal{C}$

- input: (G, X, k) where $R = G \setminus X \in \mathcal{C}$
- question: decide if $VC(G) \leq k$
- parameter: $|X|$



Landscape

New race!

Goal: obtain polynomial kernel VC|dist-to-$\mathcal{C}$ for $\mathcal{C}$ as general as possible.

VC|dist-to-$\mathcal{C}$

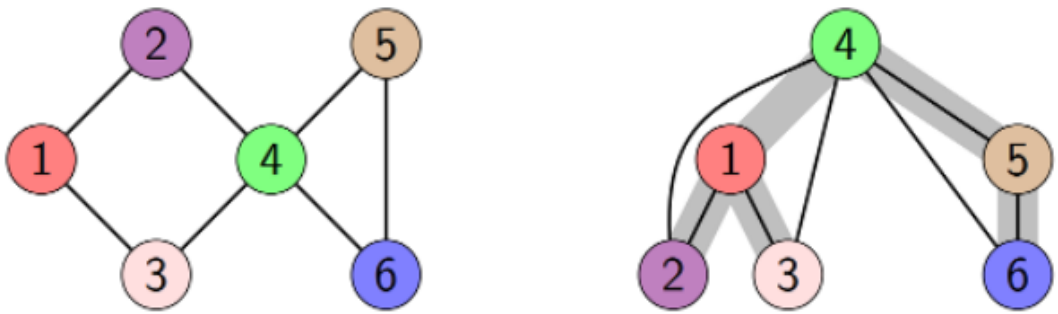
- input: (G, X, k) where $R = G \setminus X \in \mathcal{C}$
- question: decide if $VC(G) \leq k$
- parameter: $|X|$

Important examples of such class $\mathcal{C}$ are

- graphs of constant treedepth,
- graphs of constant elimination distance

Interlude: treedepth and elimination distance

A treedepth decomposition of a connected graph G is a rooted tree T on $V(G)$ such that for every edge $\{u, v\} \in E(G)$, either u is an ancestor of v or v is an ancestor of u.



The treedepth of G ($td(G)$) is the minimum depth (in number of vertices) of a treedepth decomposition of G.

Interlude: treedepth and elimination distance

A $\mathcal{B}$-elimination tree of G is a treedepth decomposition where the leaves L_i is a subset of vertices such that $G[L_i] \in \mathcal{B}$. The edges still cannot cross between different branches.

A graph G with $ed_{Forest}(G) \leq 3$.

Picture from B.M.P Jansen

The $\mathcal{B}$-elimination distance of G ($ed_{\mathcal{B}}(G)$) is the minimum depth (in number of vertices) of a $\mathcal{B}$-elimination tree of G.

Interlude: treedepth and elimination distance

A $\mathcal{B}$-elimination tree of G is a treedepth decomposition where the leaves L_i is a subset of vertices such that $G[L_i] \in \mathcal{B}$. The edges still cannot cross between different branches.

A graph G with $ed_{Forest}(G) \leq 3$.

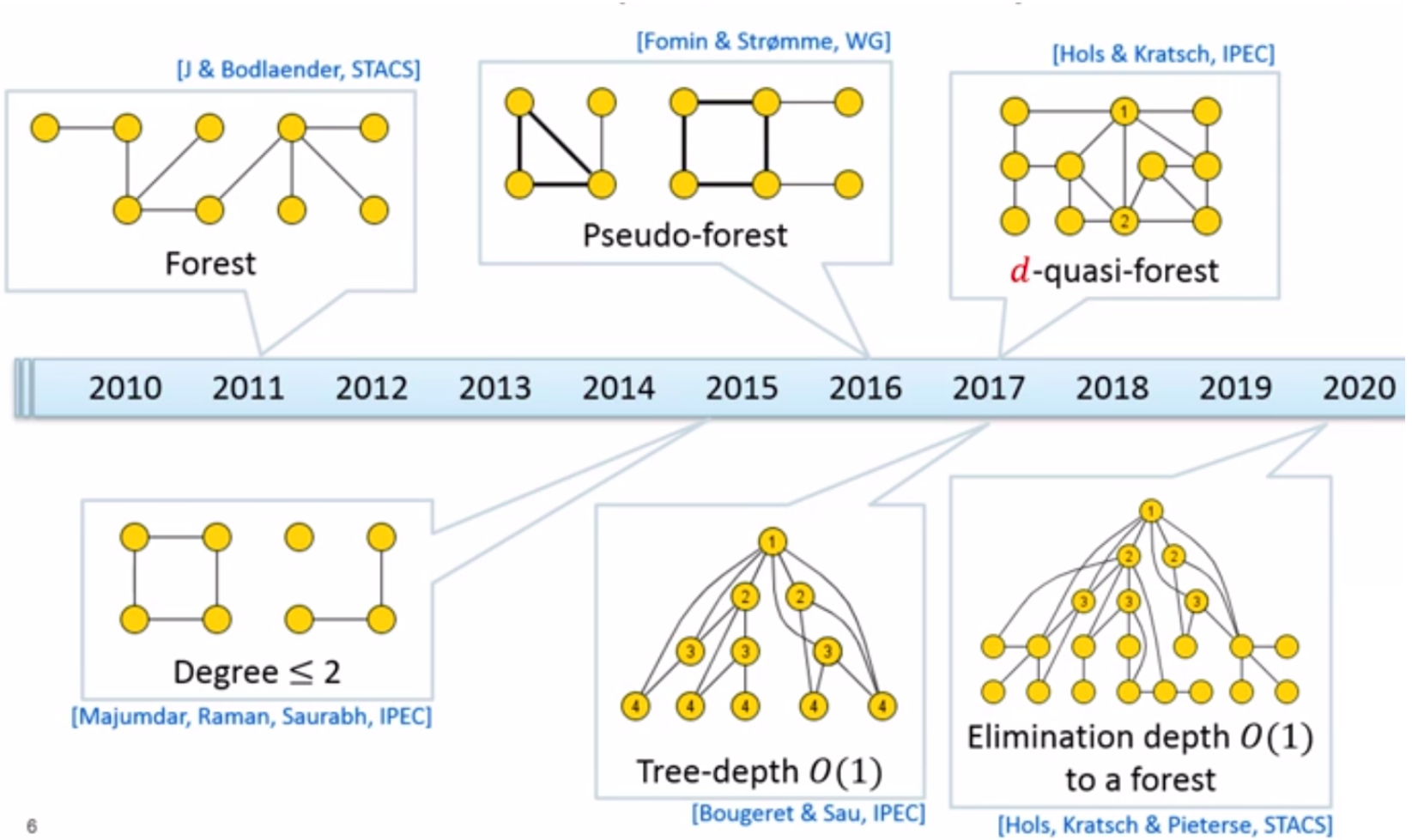
Picture from B.M.P Jansen

The $\mathcal{B}$-elimination distance of G ($ed_{\mathcal{B}}(G)$) is the minimum depth (in number of vertices) of a $\mathcal{B}$-elimination tree of G.

treedepth $\Leftrightarrow$ $\emptyset$-elimination distance.

Landscape: positive side

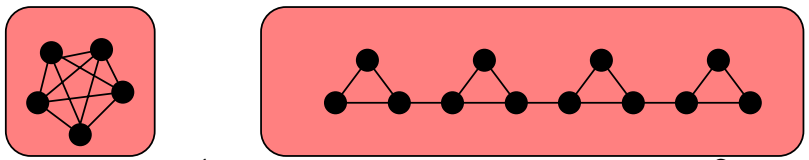
VC|dist-to-$\mathcal{C}$ has a polynomial kernel for $\mathcal{C}$ being



Picture from B.M.P Jansen

Landscape: negative side

VC|dist-to-$\mathcal{C}$ does not admit a poly kernel (unless ..) for $\mathcal{C}$ being



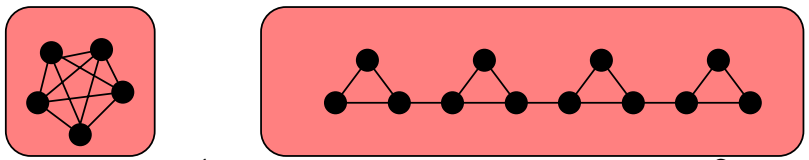
cliques[1] paths of triangles[2]

1 (Cygan, Lokshtanov, Pilipczuk x2, Saurabh)

2 (Bodlaender, Jansen, Kratsch)

Landscape: negative side

VC|dist-to-$\mathcal{C}$ does not admit a poly kernel (unless ..) for $\mathcal{C}$ being



cliques[1] paths of triangles[2]

1 (Cygan, Lokshtanov, Pilipczuk x2, Saurabh)

2 (Bodlaender, Jansen, Kratsch)

Ultimate goal: dichotomy

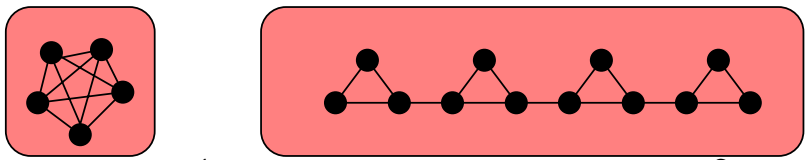
VC|dist-to-$\mathcal{C}$ admits a poly kernel iff $\mathcal{C}$ verifies ?

- Hard question (still open even for $\mathcal{C}$=bipartite graphs).

$\Rightarrow$ We restrict ourselves to minor closed families $\mathcal{C}$.

Landscape: negative side

VC|dist-to-$\mathcal{C}$ does not admit a poly kernel (unless ..) for $\mathcal{C}$ being



cliques[1] paths of triangles[2]

1 (Cygan, Lokshtanov, Pilipczuk x2, Saurabh)

2 (Bodlaender, Jansen, Kratsch)

Ultimate goal: dichotomy

VC|dist-to-$\mathcal{C}$ admits a poly kernel iff $\mathcal{C}$ verifies ?

- Hard question (still open even for $\mathcal{C}$=bipartite graphs).

$\Rightarrow$ We restrict ourselves to minor closed families $\mathcal{C}$.

Dichotomy for minor closed class $\mathcal{C}$ (Bougeret, Jansen, Sau (2020))

VC|dist-to-$\mathcal{C}$ admits a poly kernel iff $bd(\mathcal{C}) = \mathcal{O}(1)$ (bd is bridge-depth, a graph measure generalizing treedepth).

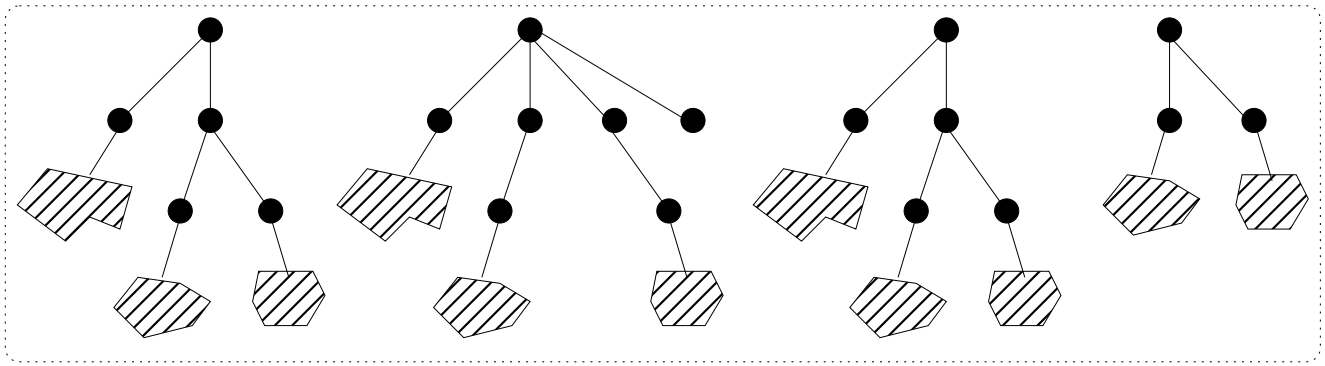
Outline

Another known dichotomy

Another known dichotomy

Dichotomy for *FVS* (Dekker, Jansen (2023))

FVS|dist-to-$\mathcal{C}$ admits a poly kernel iff $ed_{Forest}(\mathcal{C}) = \mathcal{O}(1)$.



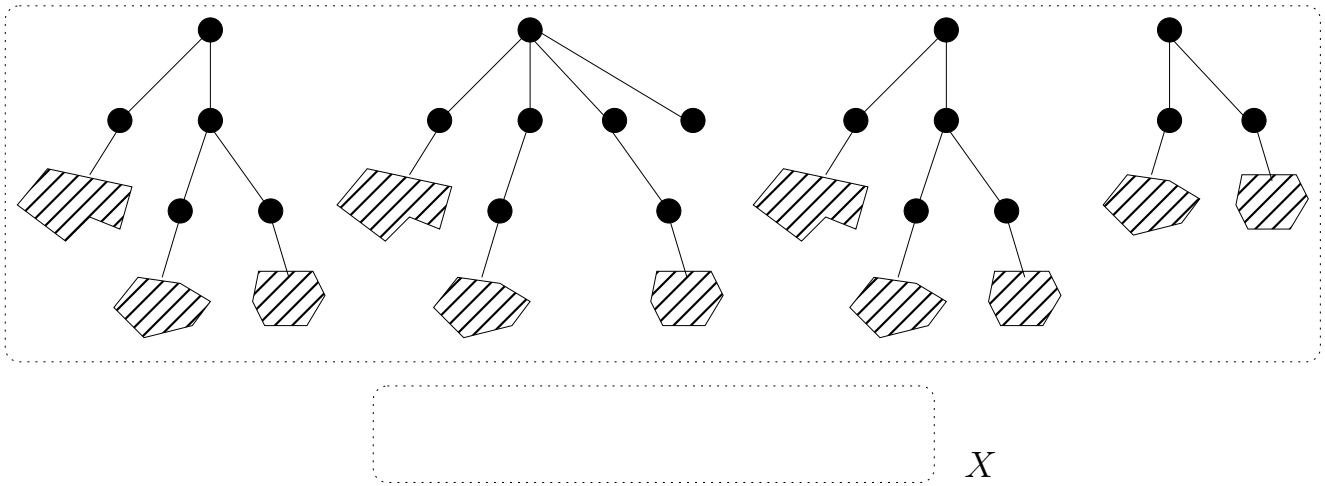
$ed_F(R) \leq c$

X

Another known dichotomy

Dichotomy for *FVS* (Dekker, Jansen (2023))

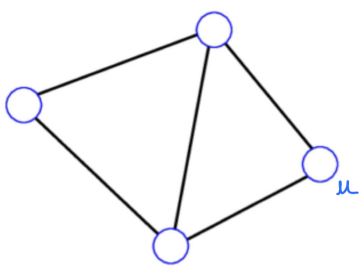
FVS|dist-to-$\mathcal{C}$ admits a poly kernel iff $ed_{Forest}(\mathcal{C}) = \mathcal{O}(1)$.



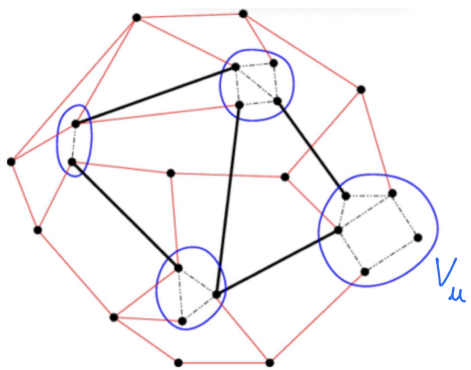
- What about other problems ?
- We generalize these results to $\mathcal{F}$-minor deletion problems as follows.

Minors

- A graph F is a minor of a graph G if for each $u \in V(F)$ you can find one connected $V_u \subseteq V(G)$, such that
  - V_u's are disjoint, and
  - for any edge $\{u_1, u_2\}$ in F there is an edge between V_{u_1} and V_{u_2} in G.
- A graph G is F-minor-free if G does not contain F as a minor.



F

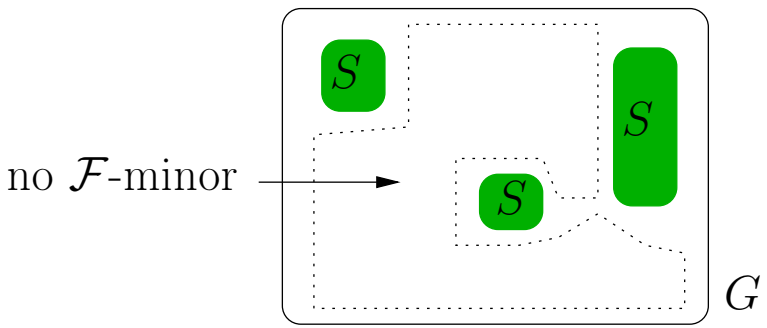


G

$\mathcal{F}$-minor deletion problem

$\mathcal{F}$-minor-deletion

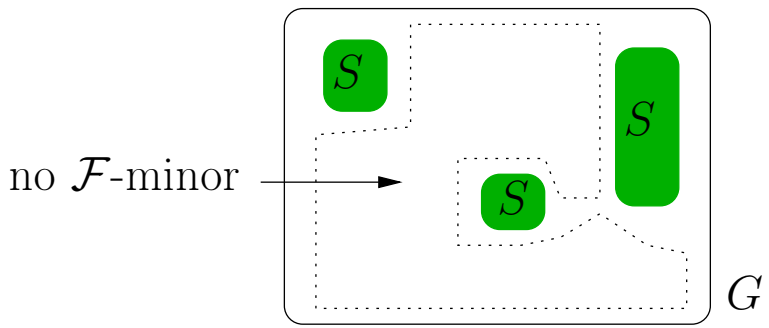
- input: (G, k)
- question: is there a set S of at most k vertices such that $G \setminus S$ is $\mathcal{F}$-minor-free.



$\mathcal{F}$-minor deletion problem

$\mathcal{F}$-minor-deletion

- input: (G, k)
- question: is there a set S of at most k vertices such that $G \setminus S$ is $\mathcal{F}$-minor-free.

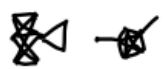


Generalizes many problems

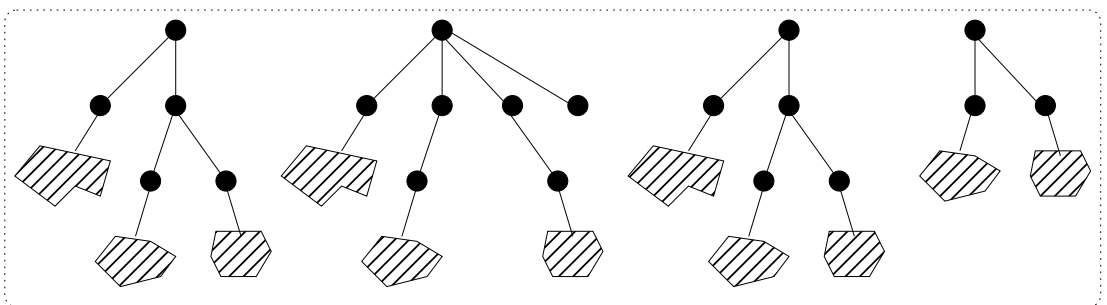
- Vertex Cover: K_2-minor deletion
- Feedback Vertex Set: K_3-minor deletion
- Planar deletion: $\mathcal{F} = \{K_5, K_{3,3}\}$-minor deletion
- ..

Kernelization of F-minor deletion: landscape

Let us denote $ed_{\mathcal{F}} = ed_{\mathcal{F}-minor-free}$

No poly kernel when	$ed_F(\mathcal{C}) = \infty$ [1]	$ed_F(\mathcal{C}) = \infty$ [1]	?
Poly kernel when			
	$F = \Delta$	$F =$  ... biconnected planar	$F =$  connected

1. (Dekker, Jansen (2023))
2. (Pieterse, Jansen (2020))
3. (Fomin, Misra, Saurabh (2012)) + (Gupta, Lee, Li, Manurangsi, Włodarczyk (2019))

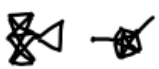


$ed_F(R) \leq c$

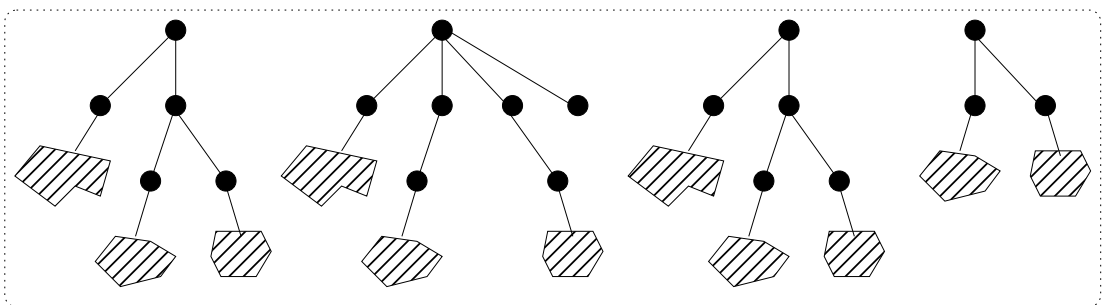
X

Kernelization of F-minor deletion: landscape

Let us denote $ed_{\mathcal{F}} = ed_{\mathcal{F}-minor-free}$

No poly kernel when	$ed_F(\mathcal{C}) = \infty$ [1]	$ed_F(\mathcal{C}) = \infty$ [1]	?
Poly kernel when	$ed_F(\mathcal{C}) = O(1)$ [1]		
	$F = \Delta$	$F =$  ... biconnected planar	$F =$  connected

1. (Dekker, Jansen (2023))
2. (Pieterse, Jansen (2020))
3. (Fomin, Misra, Saurabh (2012)) + (Gupta, Lee, Li, Manurangsi, Włodarczyk (2019))

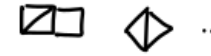
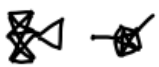


$ed_F(R) \leq c$

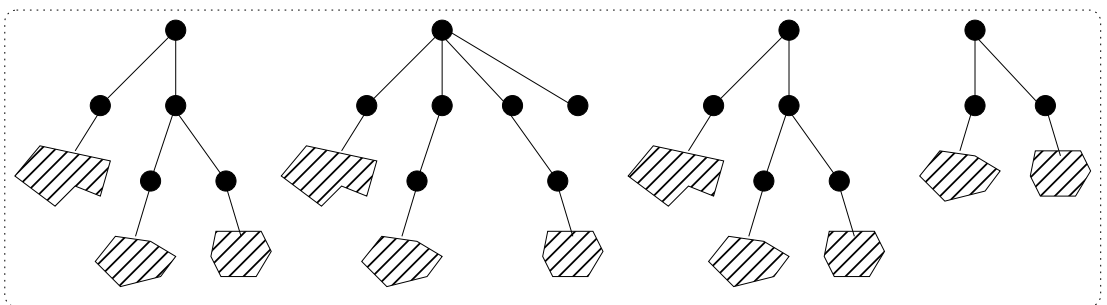
X

Kernelization of F-minor deletion: landscape

Let us denote $ed_{\mathcal{F}} = ed_{\mathcal{F}-minor-free}$

No poly kernel when	$ed_F(\mathcal{C}) = \infty$ [1]	$ed_F(\mathcal{C}) = \infty$ [1]	?
Poly kernel when	$ed_F(\mathcal{C}) = O(1)$ [1]	$td(\mathcal{C}) = O(1)$ [2]	$td(\mathcal{C}) = O(1)$ [2]
	$F = \Delta$	$F =$  biconnected planar	$F =$  connected

1 (Dekker, Jansen (2023))
2 (Pieterse, Jansen (2020))
3 (Fomin, Misra, Saurabh (2012)) + (Gupta, Lee, Li, Manurangsi, Włodarczyk (2019))

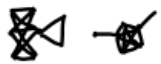


$ed_F(R) \leq c$

X

Kernelization of F-minor deletion: landscape

Let us denote $ed_{\mathcal{F}} = ed_{\mathcal{F}-minor-free}$

No poly kernel when	$ed_F(\mathcal{C}) = \infty$ [1]	$ed_F(\mathcal{C}) = \infty$ [1]	?
Poly kernel when	$ed_F(\mathcal{C}) = O(1)$ [1]	$ed_F(\mathcal{C}) = O(1)$ $\Leftarrow$ $td(\mathcal{C}) = O(1)$ [2] $\Rightarrow$ k [3]	$ed_F(\mathcal{C}) = O(1)$ $\Leftarrow$ $td(\mathcal{C}) = O(1)$ [2] $\Rightarrow$ k?
	$F = \Delta$	$F =$  ... biconnected planar	$F =$  connected

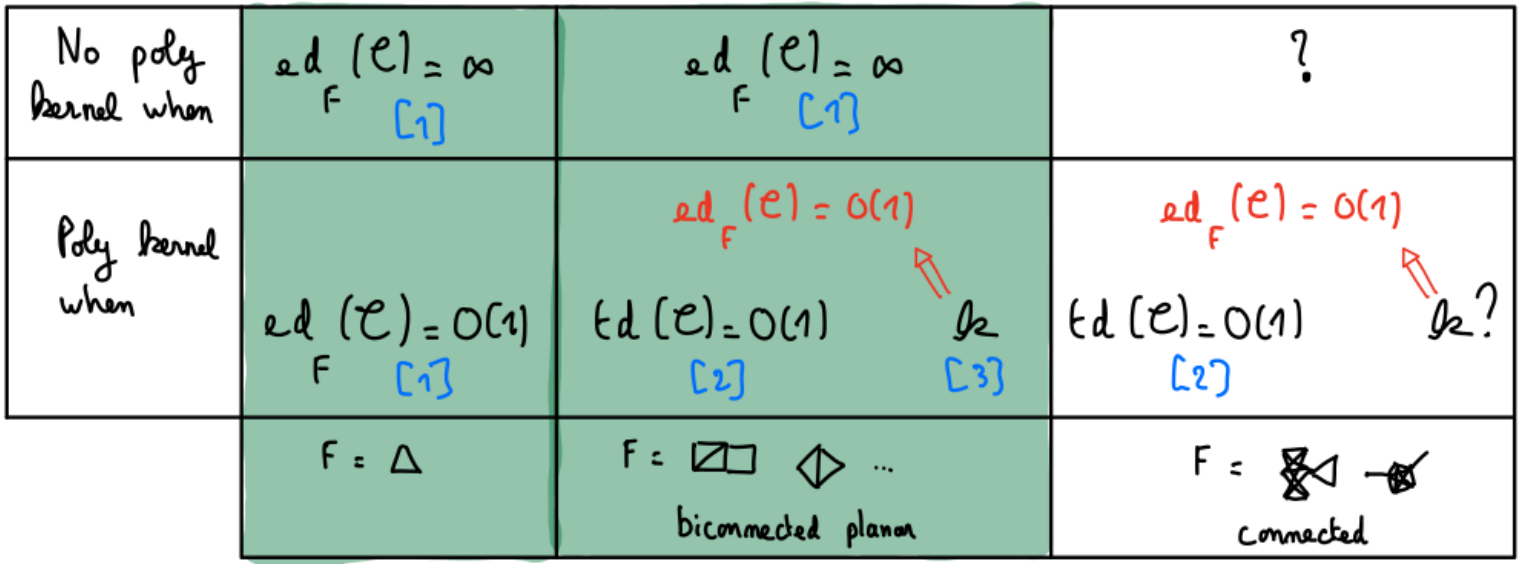
1 (Dekker, Jansen (2023))
2 (Pieterse, Jansen (2020))
3 (Fomin, Misra, Saurabh (2012)) + (Gupta, Lee, Li, Manurangsi, Włodarczyk (2019))

Our lifting result

For connected $\mathcal{F}$:
if $\mathcal{F}$-Minor Deletion$|k$ admits an (approximate) polynomial kernel, then $\mathcal{F}$-Minor Deletion|dist-to-$\mathcal{C}$ admits an (approximate) polynomial kernel for $ed_{\mathcal{F}}(\mathcal{C}) = \mathcal{O}(1)$.

Kernelization of F-minor deletion: landscape

Let us denote $ed_{\mathcal{F}} = ed_{\mathcal{F}-minor-free}$



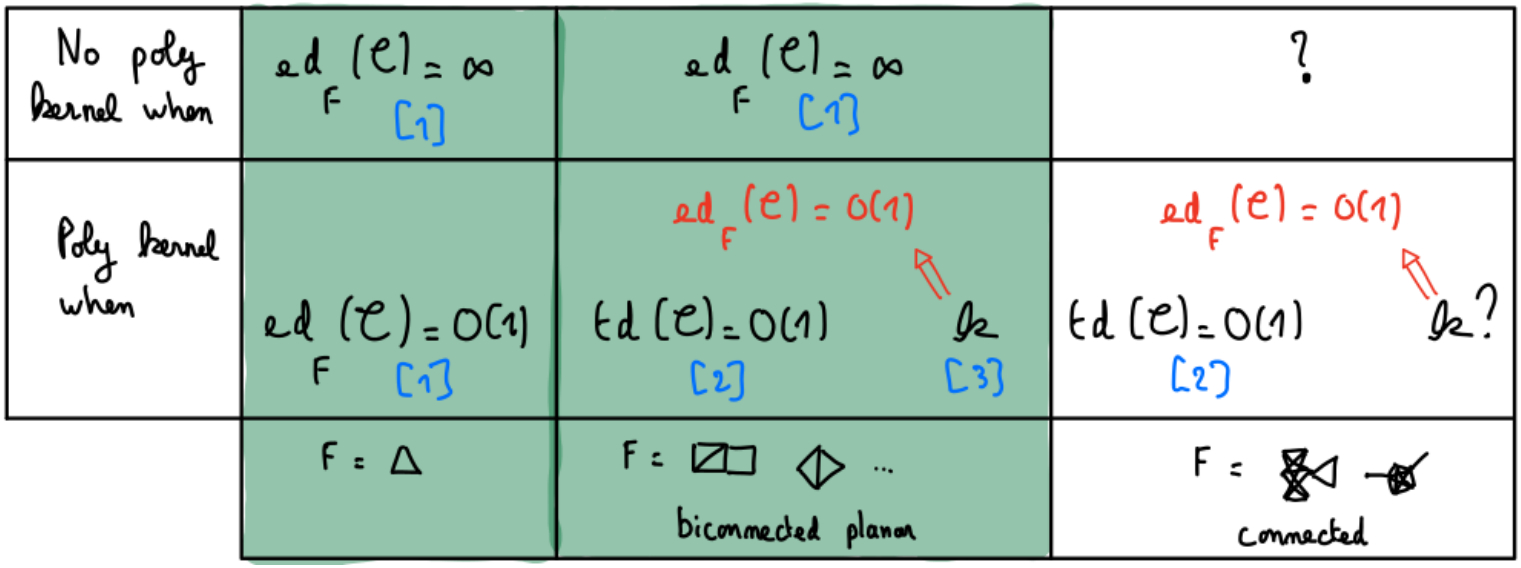
1 (Dekker, Jansen (2023))
2 (Pieterse, Jansen (2020))
3 (Fomin, Misra, Saurabh (2012)) + (Gupta, Lee, Li, Manurangsi, Włodarczyk (2019))

Corollary: new dichotomies!

For biconnected $\mathcal{F}$ with at least one planar:
$\mathcal{F}$-Minor Deletion|dist-to-$\mathcal{C}$ admits a polynomial kernel iff $ed_{\mathcal{F}}(\mathcal{C}) = \mathcal{O}(1)$.

Kernelization of F-minor deletion: landscape

Let us denote $ed_{\mathcal{F}} = ed_{\mathcal{F}-minor-free}$



1 (Dekker, Jansen (2023))
2 (Pieterse, Jansen (2020))
3 (Fomin, Misra, Saurabh (2012)) + (Gupta, Lee, Li, Manurangsi, Włodarczyk (2019))

Another recent lifting result (Jansen, de Kroon, Włodarczyk)

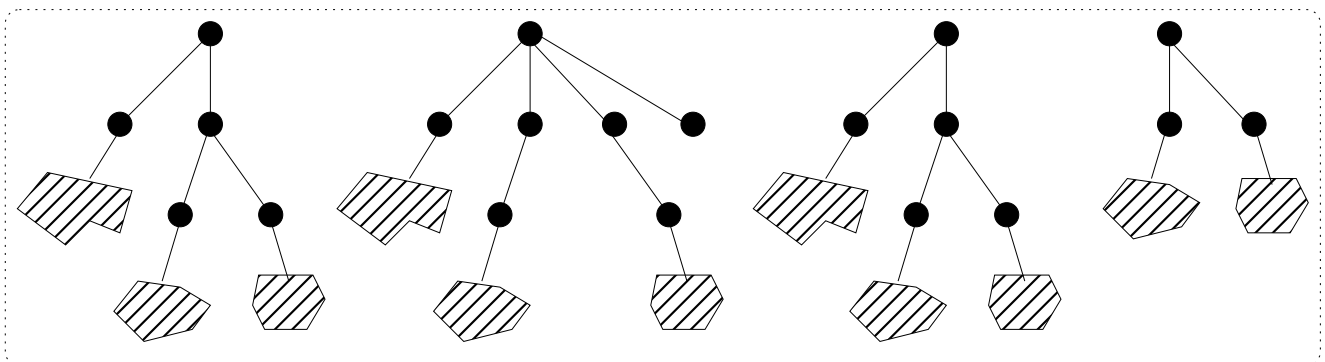
For connected $\mathcal{F}$:
if $\mathcal{F}$-Minor Deletion$|k$ is FPT,
then $\mathcal{F}$-Minor Deletion$|ed_{\mathcal{F}}$ (or $|tw_{\mathcal{F}}$) is FPT.

Outline

Overall strategy (used in many papers)

Setup for F-Minor Deletion (where F is one connected graph)

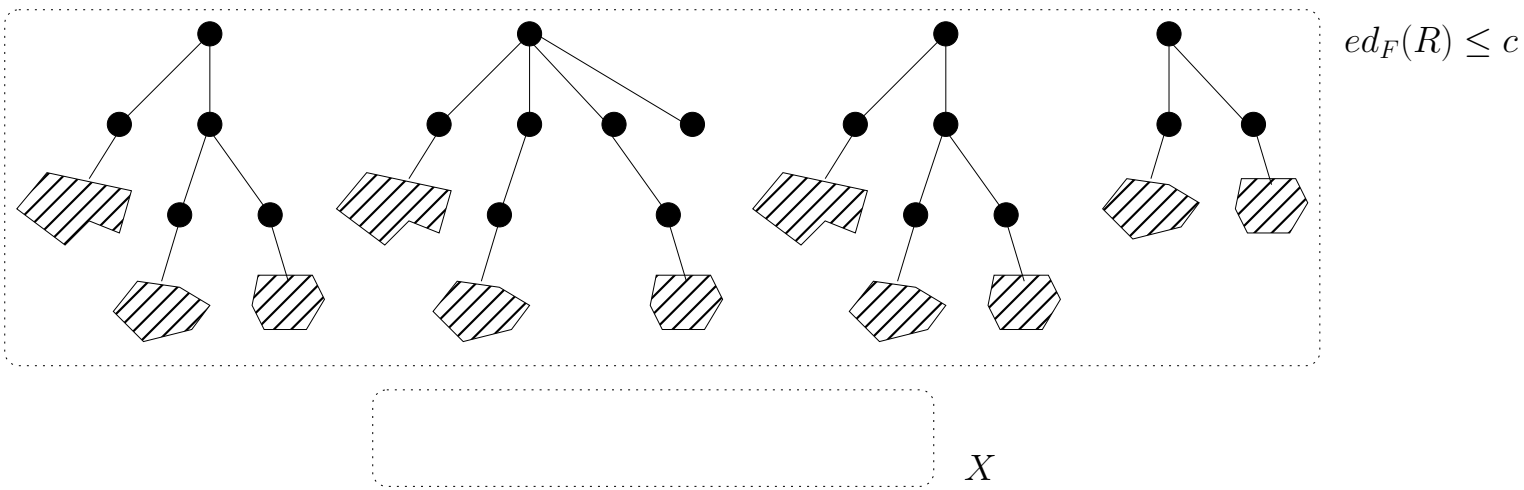
- given an instance (G, X, k), the question is to decide if there are k vertices hitting all F-minor models
- partition $V(G) = X \cup R$ with $ed_F(R) \leq c$
- goal is to reduce $V(G)$ to $X^{\mathcal{O}_c(1)}$



Overall strategy (used in many papers)

Proof scheme

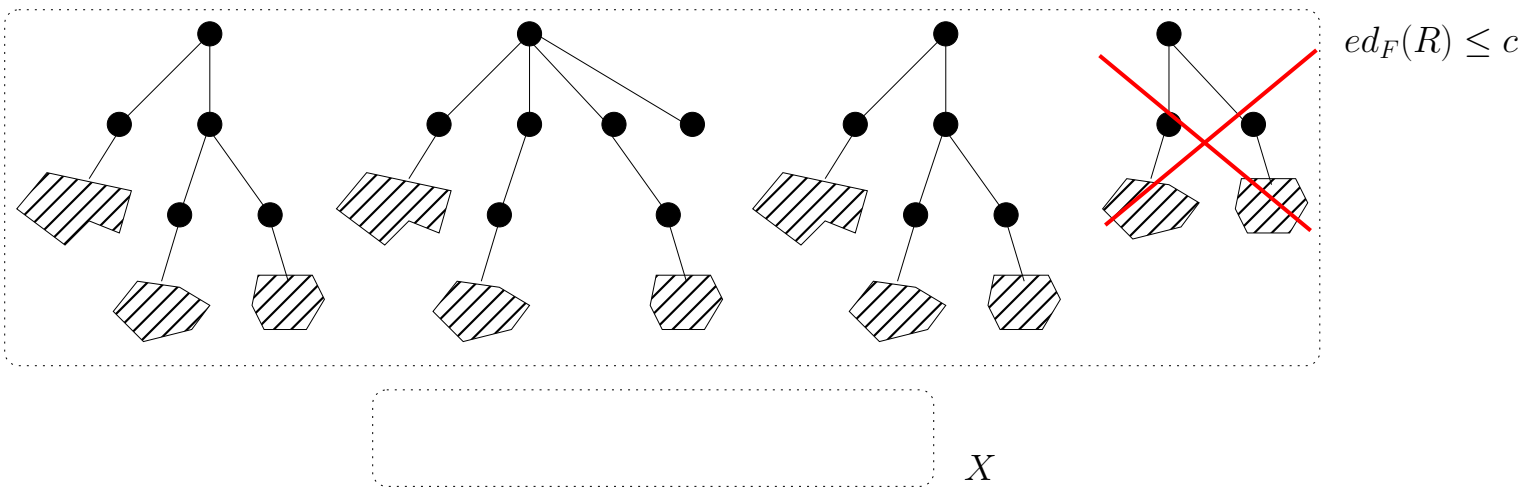
- reduce the number of cc of R to $X^{\mathcal{O}_c(1)}$
- move the root of each cc to X to get a new X', R': recurse as $ed_F(R') \leq c - 1$
- base case requires kernel by solution size (hence lifting!)



Overall strategy (used in many papers)

Proof scheme

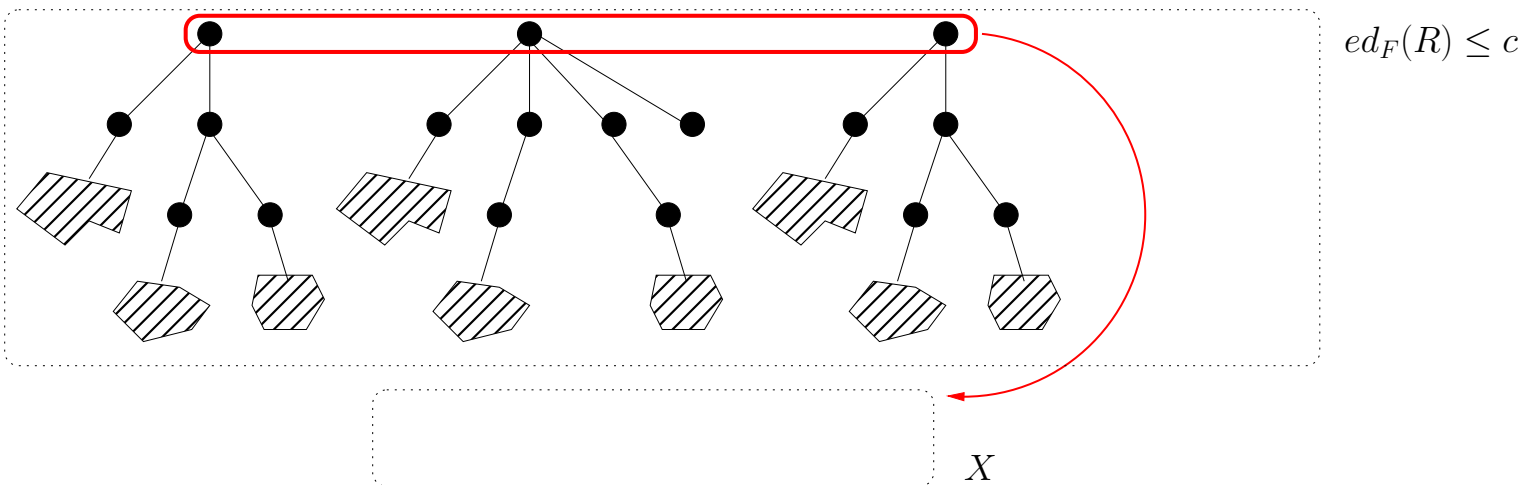
- reduce the number of cc of R to $X^{\mathcal{O}_c(1)}$
- move the root of each cc to X to get a new X', R': recurse as $ed_F(R') \leq c - 1$
- base case requires kernel by solution size (hence lifting!)



Overall strategy (used in many papers)

Proof scheme

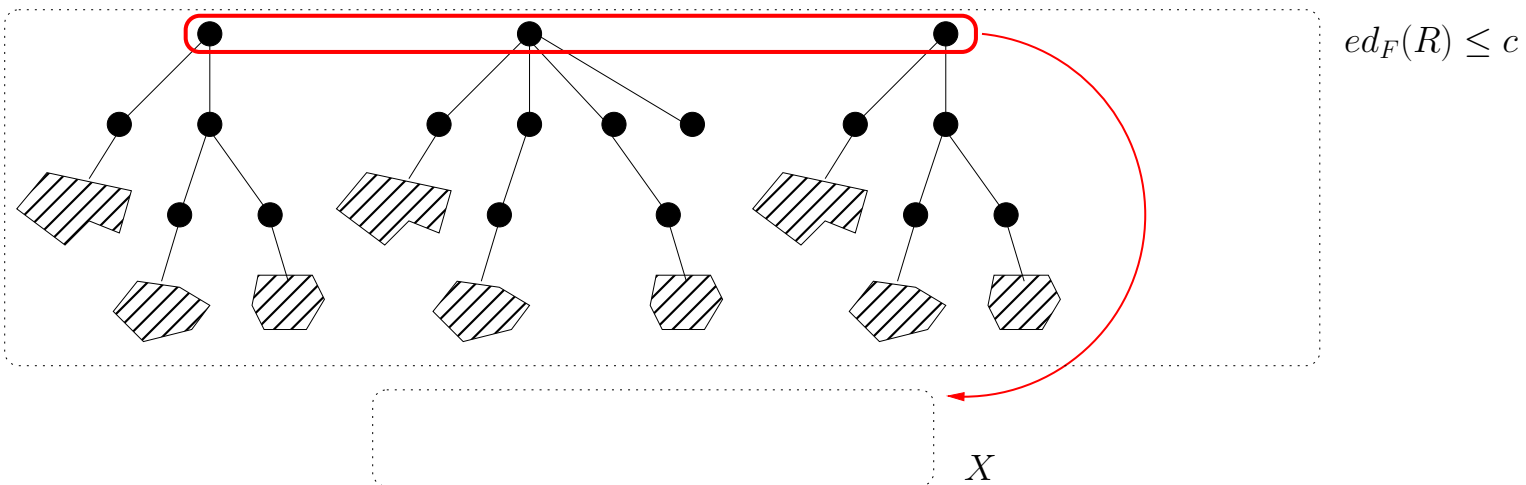
- reduce the number of cc of R to $X^{\mathcal{O}_c(1)}$
- move the root of each cc to X to get a new X', R': recurse as $ed_F(R') \leq c - 1$
- base case requires kernel by solution size (hence lifting!)



Overall strategy (used in many papers)

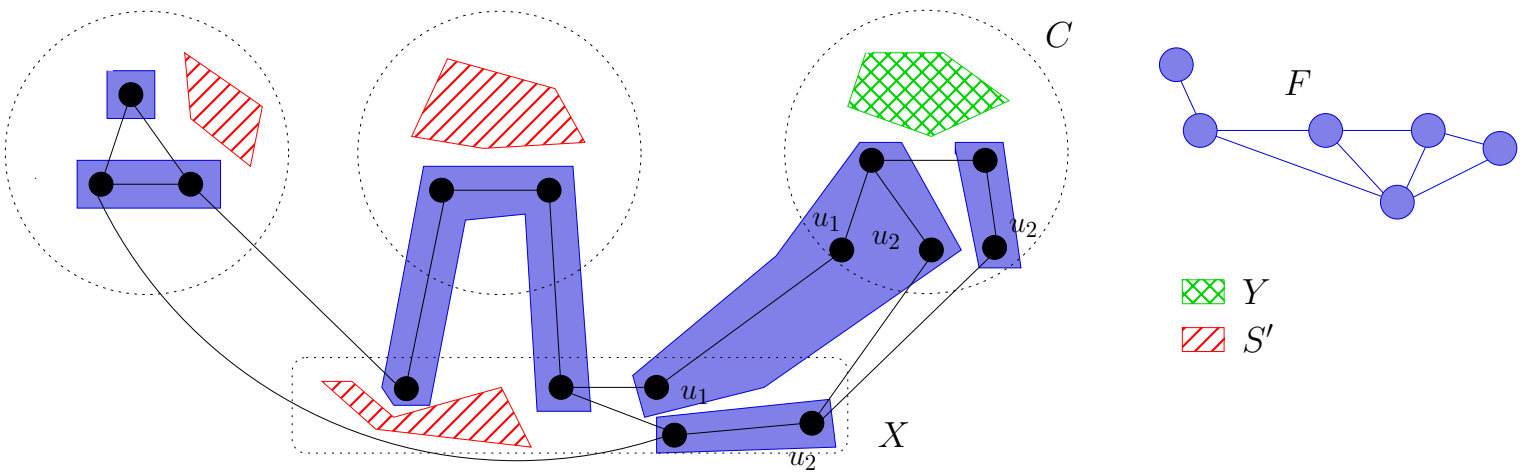
Proof scheme

- reduce the number of cc of R to $X^{\mathcal{O}_c(1)}$
- move the root of each cc to X to get a new X', R': recurse as $ed_F(R') \leq c - 1$
- base case requires kernel by solution size (hence lifting!)



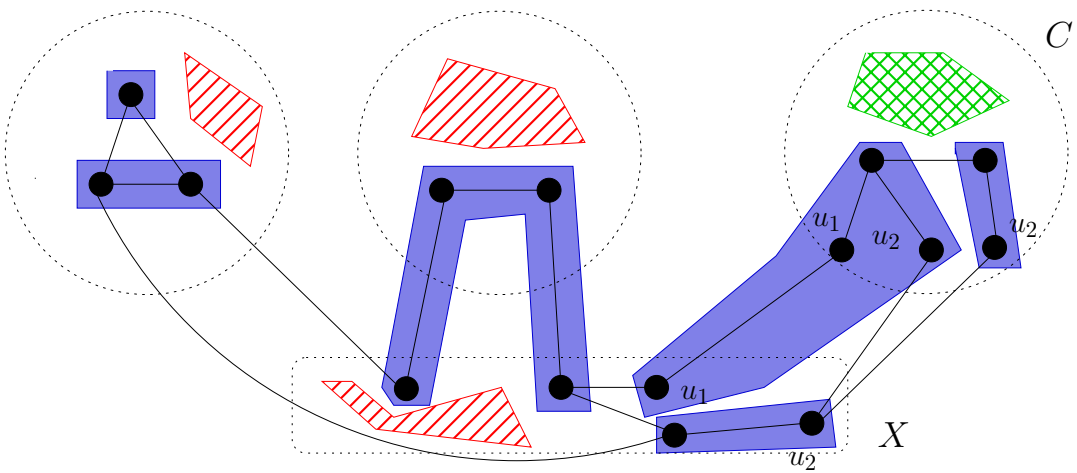
How to remove a connected component

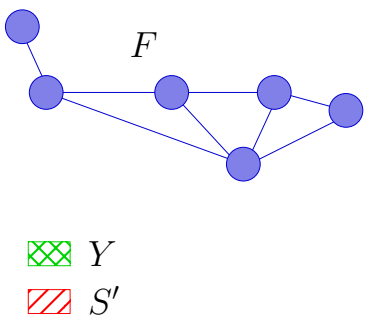
- let us say we want to remove C, and update $k' = k - opt(C)$
- how to ensure that (G', k') yes $\Rightarrow$ (G, k) yes ?
- let S' be solution of (G', k'), and Y be an optimal of $G[C]$
- danger: $S' \cup Y$ may not hit all F-minors of G



How to remove a connected component

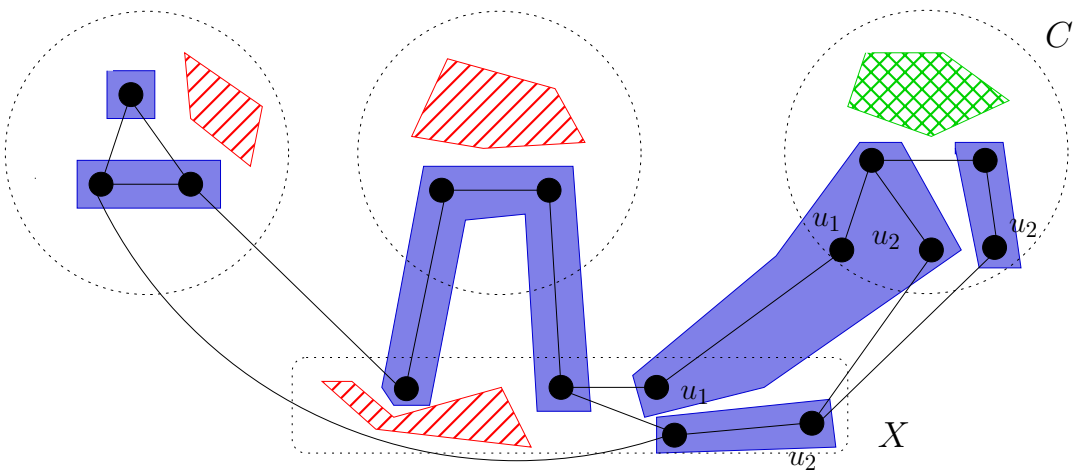
- let us say we want to remove C, and update $k' = k - opt(C)$
- how to ensure that (G', k') yes $\Rightarrow$ (G, k) yes ?
- let S' be solution of (G', k'), and Y be an optimal of $G[C]$
- danger: $S' \cup Y$ may not hit all F-minors of G

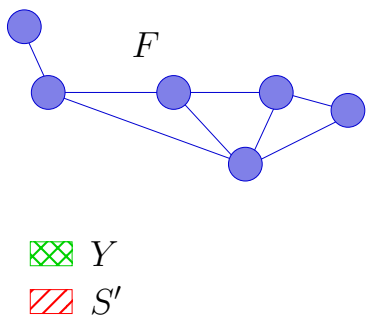




How to remove a connected component

- let us say we want to remove C, and update $k' = k - opt(C)$
- how to ensure that (G', k') yes $\Rightarrow$ (G, k) yes ?
- let S' be solution of (G', k'), and Y be an optimal of $G[C]$
- danger: $S' \cup Y$ may not hit all F-minors of G

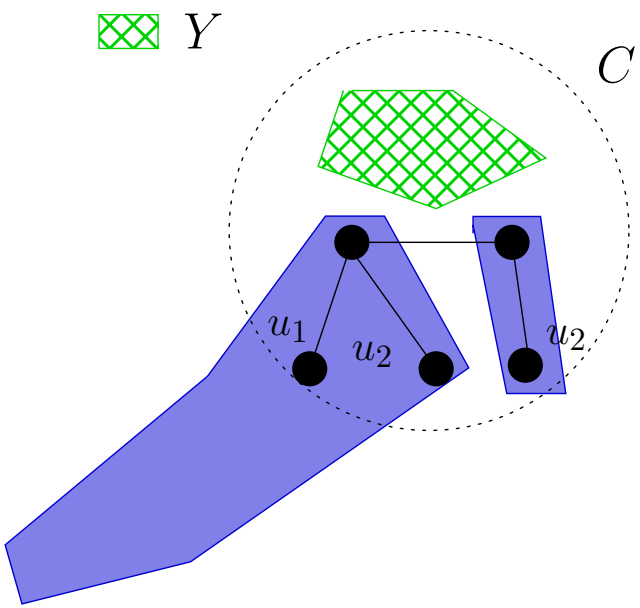




How to remove a connected component

We would like a magic local optimal Y of $G[C]$ that also kills:

- all fragments in $Q_{S'}$..
- and all fragments in $Q_{S'_2}$..
- ..

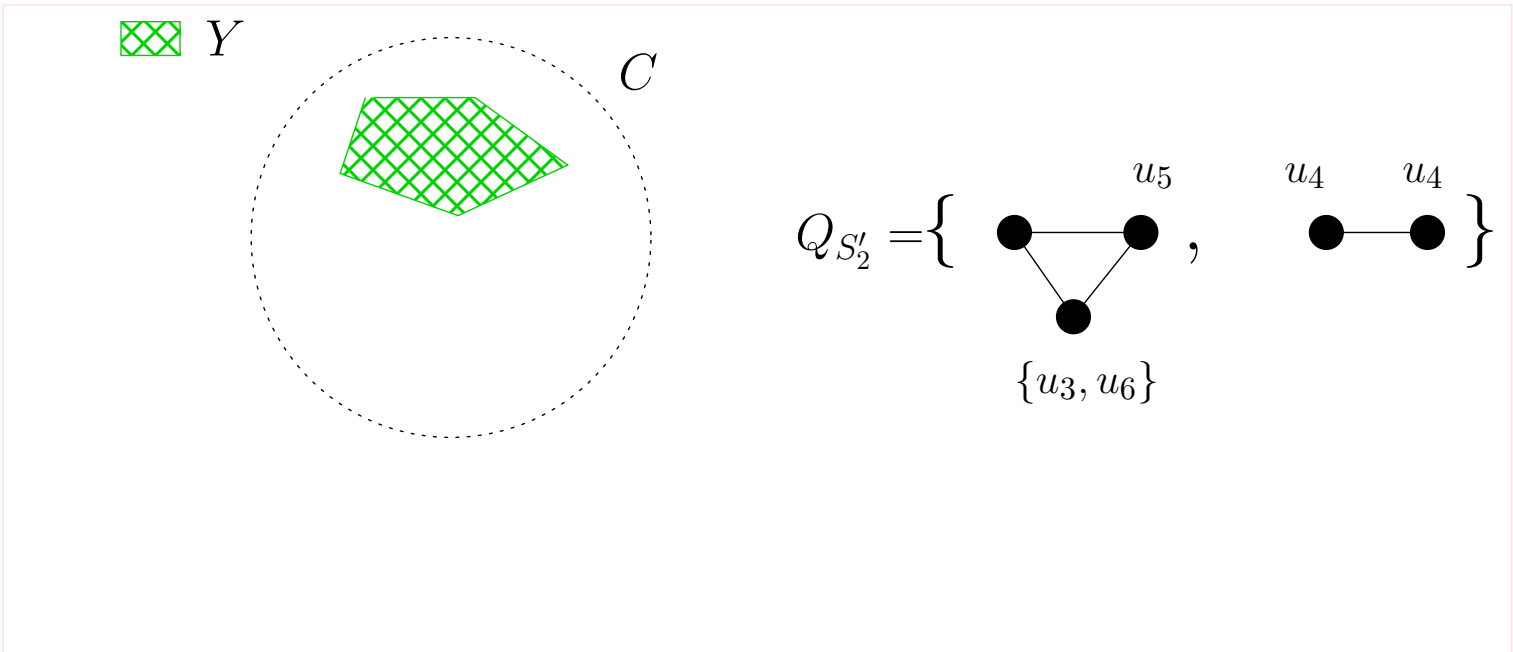


$$Q_{S'} = \{ \overset{\{u_1,u_2\}}{\bullet\!\!-\!\!\bullet} \quad \overset{u_2}{\bullet} \}$$

How to remove a connected component

We would like a magic local optimal Y of $G[C]$ that also kills:

- all fragments in $Q_{S'}$..
- and all fragments in $Q_{S'_2}$..
- ..

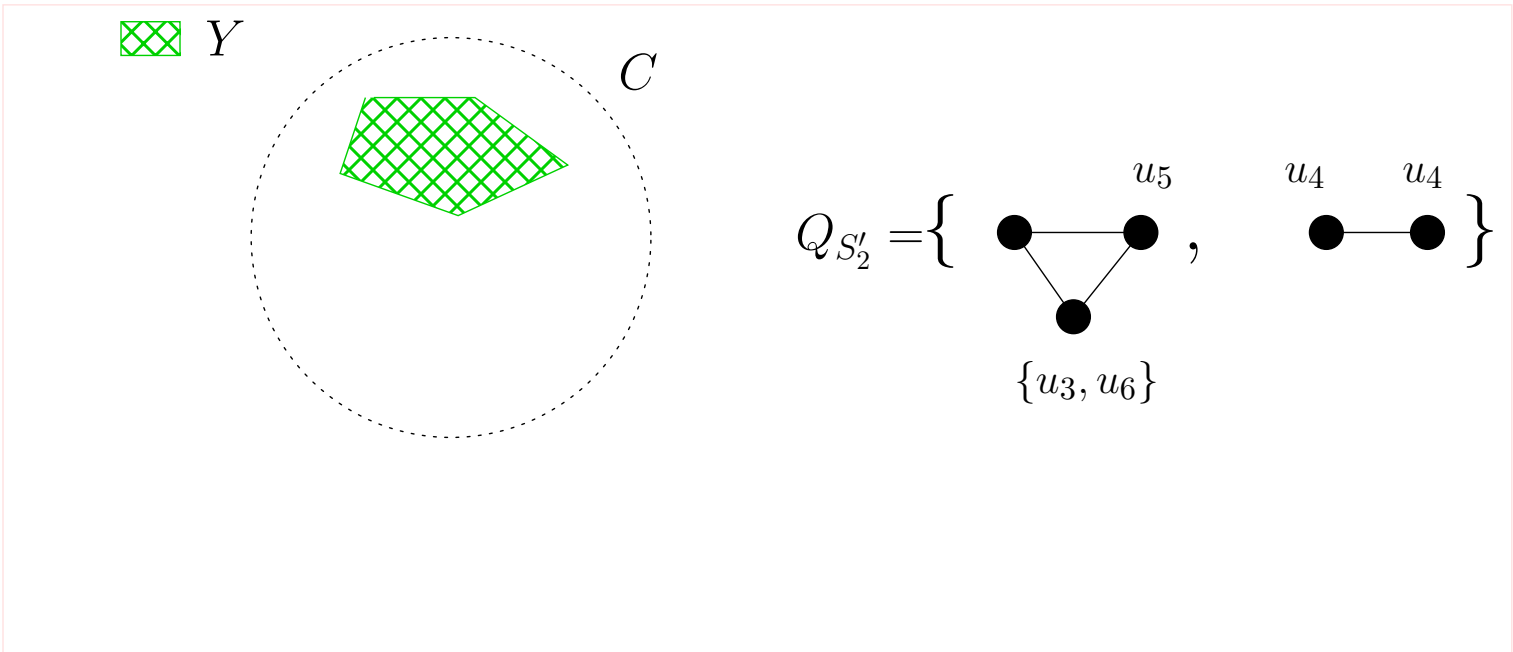


How to remove a connected component

We would like a magic local optimal Y of $G[C]$ that also kills:

- all fragments in $Q_{S'}$..
- and all fragments in $Q_{S'_2}$..
- ..

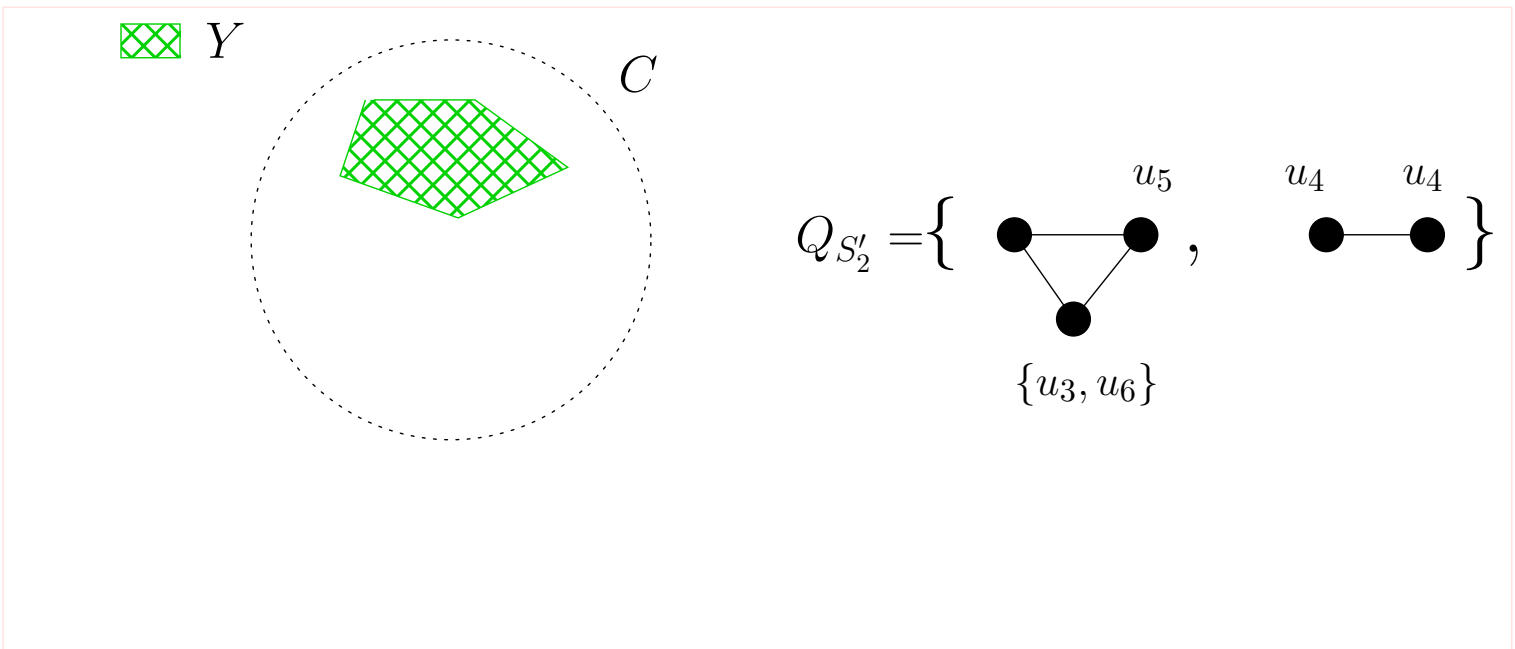
Problem: number of possible $Q_{..}$ is not polynomial !



How to remove a connected component

Blocking set lemma (informal statement, generalized version of (Jansen, Pieterse (2020)))

Let Q be a set X-labeled fragment of F, and let C be an X-labeled graph. If all optimal solutions to F-Minor Deletion on C leave a Q-minor, then there is a subset $Q^\star \subseteq Q$ whose size depends only on $(F, ed_F(C))$, such that all optimal solutions leave a $Q^\star - minor$.

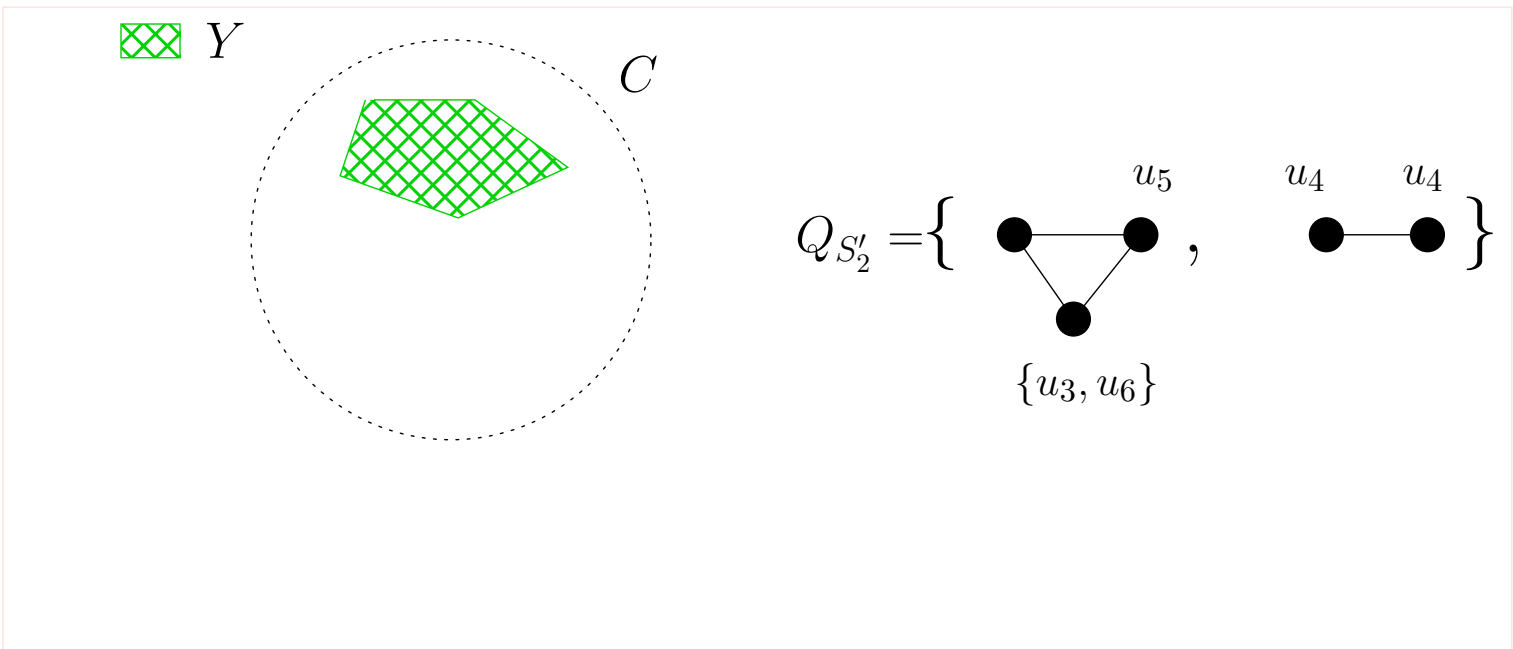


How to remove a connected component

Blocking set lemma (informal statement, generalized version of (Jansen, Pieterse (2020)))

Let Q be a set X-labeled fragment of F, and let C be an X-labeled graph. If all optimal solutions to F-Minor Deletion on C leave a Q-minor, then there is a subset $Q^\star \subseteq Q$ whose size depends only on $(F, ed_F(C))$, such that all optimal solutions leave a $Q^\star - minor$.

We can now enumerate all possible $Q^\star_{..}$ in polynomial time !

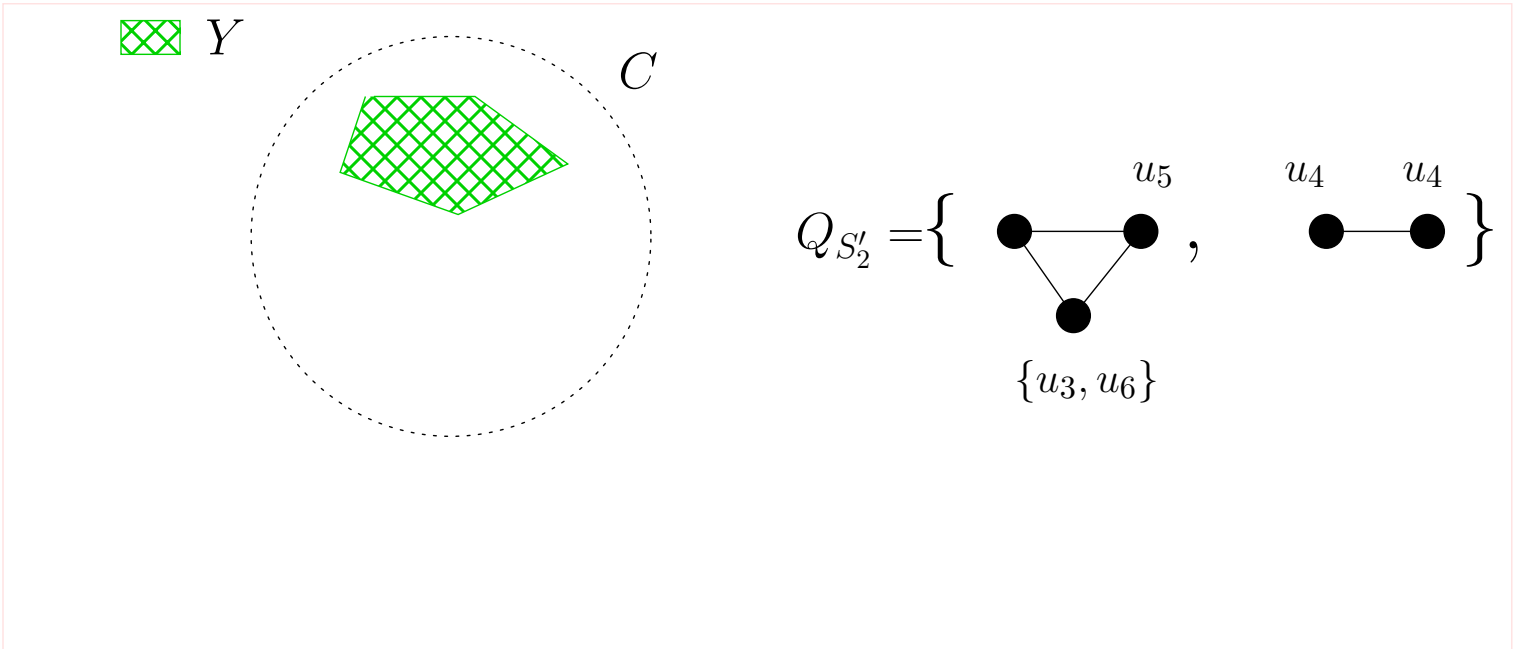


How to remove a connected component

We would like a magic local optimal Y of $G[C]$ that also kills:

- all fragments in $Q_{S'} \leftrightarrow$ all fragments in $Q^{\star}_{S'}$
- and all fragments in $Q_{S'_2} \leftrightarrow$ all fragments in $Q^{\star}_{S'_2}$
- ..

We can now enumerate all possible $Q^{\star}_{..}$ in polynomial time !



Conclusions

The kernel algorithm is the same as in (Jansen, Pieterse (2020)) (which considers $td(\mathcal{C}) = \mathcal{O}(1)$), up to two ingredients.

Ingredient 1: blocking set lemma

- Blocking set lemma was proved in (Jansen, Pieterse (2020)) for $td(\mathcal{C}) = \mathcal{O}(1)$ (instead of $ed_F(\mathcal{C}) = \mathcal{O}(1)$).
- We re-use the consequent theory on labeled minors and very technical inductive proof of (Jansen, Pieterse (2020)).
- Crucial difference: in the base case of this induction, handle $\mathcal{F}$-minor graph instead of empty graph.

Conclusions

The kernel algorithm is the same as in (Jansen, Pieterse (2020)) (which considers $td(\mathcal{C}) = \mathcal{O}(1)$), up to two ingredients.

Ingredient 1: blocking set lemma

- Blocking set lemma was proved in (Jansen, Pieterse (2020)) for $td(\mathcal{C}) = \mathcal{O}(1)$ (instead of $ed_F(\mathcal{C}) = \mathcal{O}(1)$).
- We re-use the consequent theory on labeled minors and very technical inductive proof of (Jansen, Pieterse (2020)).
- Crucial difference: in the base case of this induction, handle $\mathcal{F}$-minor graph instead of empty graph.

Conclusions

The kernel algorithm is the same as in (Jansen, Pieterse (2020)) (which considers $td(\mathcal{C}) = \mathcal{O}(1)$), up to two ingredients.

Ingredient 1: blocking set lemma

- Blocking set lemma was proved in (Jansen, Pieterse (2020)) for $td(\mathcal{C}) = \mathcal{O}(1)$ (instead of $ed_F(\mathcal{C}) = \mathcal{O}(1)$).
- We re-use the consequent theory on labeled minors and very technical inductive proof of (Jansen, Pieterse (2020)).
- Crucial difference: in the base case of this induction, handle $\mathcal{F}$-minor graph instead of empty graph.

Conclusions

The kernel algorithm is the same as in (Jansen, Pieterse (2020)) (which considers $td(\mathcal{C}) = \mathcal{O}(1)$), up to two ingredients.

Ingredient 1: blocking set lemma

- Blocking set lemma was proved in (Jansen, Pieterse (2020)) for $td(\mathcal{C}) = \mathcal{O}(1)$ (instead of $ed_F(\mathcal{C}) = \mathcal{O}(1)$).
- We re-use the consequent theory on labeled minors and very technical inductive proof of (Jansen, Pieterse (2020)).
- Crucial difference: in the base case of this induction, handle $\mathcal{F}$-minor graph instead of empty graph.

Conclusions

The kernel algorithm is the same as in (Jansen, Pieterse (2020)) (which considers $td(\mathcal{C}) = \mathcal{O}(1)$), up to two ingredients.

Ingredient 2: algorithm to check that a magic opt exists

Given a set of label fragments $\mathcal{Q}$ and a labeled graph C, we need to determine if there exists an optimal (of F-minor deletion) of C that also hits all $Q \in \mathcal{Q}$.

- Ingredient 2 also required in (Jansen, Pieterse (2020)), but where $td(C) = \mathcal{O}(1)$
- More difficult here, as $ed_F(C) = \mathcal{O}(1) \not\Rightarrow tw(C) = \mathcal{O}(1)$.
- We reduce to unlabeled version and use the algorithm of (Jansen, de Kroon, Włodarczyk) for F-Minor Deletion/H_F-treewidth

Conclusions

The kernel algorithm is the same as in (Jansen, Pieterse (2020)) (which considers $td(\mathcal{C}) = \mathcal{O}(1)$), up to two ingredients.

Ingredient 2: algorithm to check that a magic opt exists

Given a set of label fragments $\mathcal{Q}$ and a labeled graph C, we need to determine if there exists an optimal (of F-minor deletion) of C that also hits all $Q \in \mathcal{Q}$.

- Ingredient 2 also required in (Jansen, Pieterse (2020)), but where $td(C) = \mathcal{O}(1)$
- More difficult here, as $ed_F(C) = \mathcal{O}(1) \not\Rightarrow tw(C) = \mathcal{O}(1)$.
- We reduce to unlabeled version and use the algorithm of (Jansen, de Kroon, Włodarczyk) for F-Minor Deletion/H_F-treewidth

Conclusions

Future work

Find some dichotomies for non biconnected F. Outside biconnected F, landscape looks erratic (on going work):

- for $F = K_2$, answer was bd, and not $ed_F (= td)$
- for $F = K_{1,3}$, answer seems to be .. $ed_{\text{caterpillar}}$, and not ed_F
- ..

Conclusions

Future work

Find some dichotomies for non biconnected F. Outside biconnected F, landscape looks erratic (on going work):

- for $F = K_2$, answer was bd, and not $ed_F (= td)$
- for $F = K_{1,3}$, answer seems to be .. $ed_{\text{caterpillar}}$, and not ed_F
- ..

Thank you for your attention !