A single exponential FPT algorithm for the Planar-$\mathcal{F}$-Deletion problem

Christophe Paul
CNRS - LIRMM, Montpellier, France

Joint work with
EunJung Kim (CNRS - LAMSADE, Paris, France)
Ignasi Sau (CNRS - LIRMM, Montpellier, France)

Nyborg Graph Theory Workshop
November 3, 2012

[E.J. Kim, A. Langer, C. Paul, P. Rossmanith, I. Sau, F. Reidl, S. Sikdar. *Linear Kernels and single exponential algorithms via protrusion decomposition.* arXiv:1207.0825, 2012]

$\mathcal{F}$-Deletion Problem

Given a graph $G = (V, E)$ and an integer k as parameter,

- is there a subset $X \subseteq V$ such that $G - X$ is $\mathcal{F}$-minor free ?

Observations :

1. $\{K_2\}$-Deletion $\equiv$ Vertex Cover
   $\equiv$ Treewidth-zero Vertex Deletion
2. $\{K_3\}$-Deletion $\equiv$ Feedback Vertex Set
   $\equiv$ Treewidth-one Vertex Deletion
3. $\{K_4\}$-Deletion $\equiv$ Treewidth-two Vertex Deletion

More generally, how fast can we solve

- Treewidth-t Vertex Deletion ?
- Planar-$\mathcal{F}$-Deletion ($\mathcal{F}$ contains a planar graph) ?

Known results (1)

When $\mathcal{F}$ is "non-planar"

- $\mathcal{F}$-Deletion is FPT
  (by the Roberston and Seymour' graph minor theorem)
- $\{K_5, K_{3,3}\}$-Deletion can be solved in $O^*(2^{2^{(k \log k)}})$
  [Marx, Schlotter'07] [Kawarabayashi'09]

When $\mathcal{F}$ is planar

$\{K_2\}$-Deletion (VC)	$O^*(1.2738^k)$	[J. Chen et al.'10]
$\{K_3\}$-Deletion (FVS)	$O^*(3.83^k)$	[Y. Cao et al',10],
$\{\theta_c\}$-Deletion	$O^*(c^k)$	[G. Joret et al.'11]
$\{K_4\}$-Deletion	$O^*(c^k)$	[E.J.Kim et al.'12]

Known results (2)

When $\mathcal{F}$ is planar (cont'd)

- $2^{2^{O(k \log k)}} \cdot n^{O(1)}$ -time algorithm based on DP
- [Fomin et al.'11] $2^{O(k \log k)} \cdot n^2$ -time algorihm
- [Fomin et al.'12] $2^{O(k)} \cdot n \log^2 n$ -time algorithm for Connected-Planar-$\mathcal{F}$-Deletion

Known results (2)

When $\mathcal{F}$ is planar (cont'd)

- $2^{2^{O(k \log k)}} \cdot n^{O(1)}$ -time algorithm based on DP
- [Fomin et al.'11] $2^{O(k \log k)} \cdot n^2$ -time algorihm
- [Fomin et al.'12] $2^{O(k)} \cdot n \log^2 n$ -time algorithm for Connected-Planar-$\mathcal{F}$-Deletion

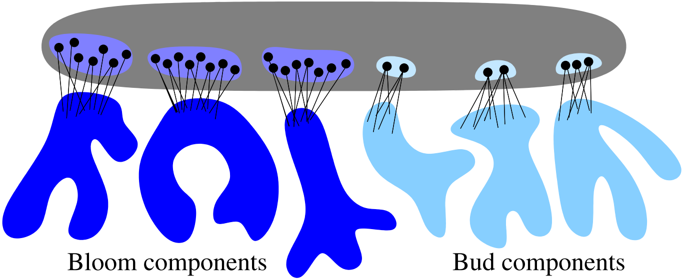
Our result:

A $2^{O(k)}.n^2$ -time algorithm for Planar-$\mathcal{F}$-Deletion

- [Chen et al.'05] No hope for a $2^{o(k)} \cdot n^{O(1)}$-time algorithm

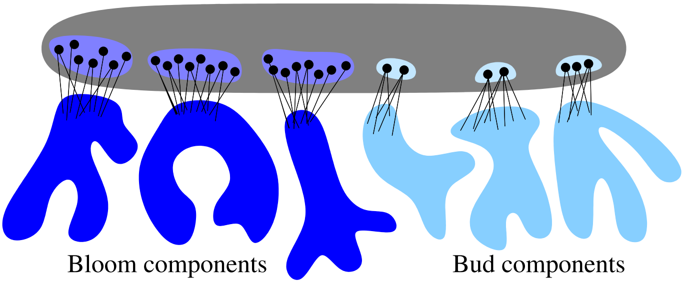
Using iterative compression the PLANAR-$\mathcal{F}$-DELETION can be reduced in single exponential time to :

DISJOINT PLANAR-$\mathcal{F}$-DELETION problem:
Given $X \subseteq V(G)$ such that $G - X$ is $\mathcal{F}$-minor free, compute $\tilde{X} \subseteq V \setminus X$ such that $G - \tilde{X}$ is $\mathcal{F}$-minor free and $|\tilde{X}| \leqslant |X| = k$?



Using iterative compression the PLANAR-$\mathcal{F}$-DELETION can be reduced in single exponential time to :

DISJOINT PLANAR-$\mathcal{F}$-DELETION problem:
Given $X \subseteq V(G)$ such that $G - X$ is $\mathcal{F}$-minor free, compute $\tilde{X} \subseteq V \setminus X$ such that $G - \tilde{X}$ is $\mathcal{F}$-minor free and $|\tilde{X}| \leqslant |X| = k$?



Working Hypothesis : An alternative solution $\tilde{X}$ does exist !

1. Guess a subset I of the alternative solution $\tilde{X}$ such that:
   - $G - I$ has a linear protrusion decomposition

     $$\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$$

   - with $X \subseteq Y_0$ and $\tilde{X} \setminus I \subseteq V \setminus Y_0$

Road map - Overview of the algorithm

1. Guess a subset I of the alternative solution $\tilde{X}$ such that:
   - $G - I$ has a linear protrusion decomposition
     $$\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$$
   - with $X \subseteq Y_0$ and $\tilde{X} \setminus I \subseteq V \setminus Y_0$

2. Compute $\tilde{X} \setminus I$ using branching on Annotated & Colored versions of the problem.

Every step runs in $O^*(c^k)$

Road map - Overview of the algorithm

1. Guess a subset I of the alternative solution $\tilde{X}$ such that:
   - $G - I$ has a linear protrusion decomposition
   $$\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$$
   - with $X \subseteq Y_0$ and $\tilde{X} \setminus I \subseteq V \setminus Y_0$

Ideas and techniques inspired from [Langer, Reidl, Rossmanith, Sikdar'12]

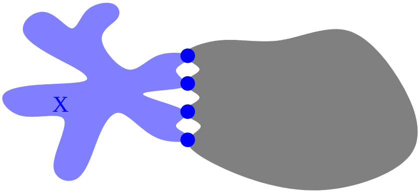
2. Compute $\tilde{X} \setminus I$ using branching on Annotated & Colored versions of the problem.

Based on annotated reduction Lemma from [Bodlaender *et al.*'09]

Every step runs in $O^*(c^k)$

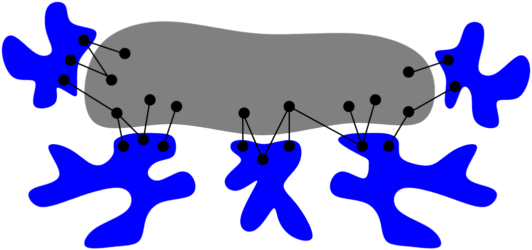
A β-protrusion in a graph G is a subset $Y \subseteq V(G)$ such that

$$|\partial(Y)| \leqslant \beta \text{ and } \mathbf{tw}(G[Y]) \leqslant \beta$$



A β-protrusion in a graph G is a subset $Y \subseteq V(G)$ such that

$$|\partial(Y)| \leqslant \beta \text{ and } \mathbf{tw}(G[Y]) \leqslant \beta$$



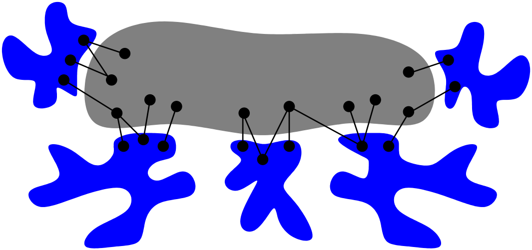
A partition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ with $\max\{\ell, |Y_0|\} \leqslant \alpha$ is an (α, β)-protrusion decomposition if for every $1 \leqslant i \leqslant \ell$,

$$N(Y_i) \subseteq Y_0 \quad \text{and} \quad Y_i \cup N_{Y_0}(Y_1) \text{ is a } \beta\text{-protrusion}$$

Linear protrusion decomposition - Definition

A β-protrusion in a graph G is a subset $Y \subseteq V(G)$ such that

$$|\partial(Y)| \leqslant \beta \text{ and } \mathbf{tw}(G[Y]) \leqslant \beta$$



A partition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ with $\max\{\ell, |Y_0|\} \leqslant \alpha$ is an (α, β)-protrusion decomposition if for every $1 \leqslant i \leqslant \ell$,

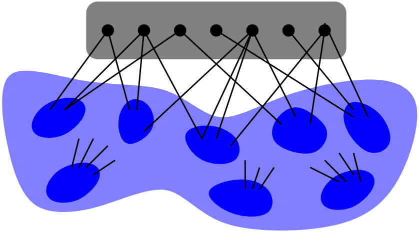
$$N(Y_i) \subseteq Y_0 \quad \text{and} \quad Y_i \cup N_{Y_0}(Y_1) \text{ is a } \beta\text{-protrusion}$$

- $\mathcal{P}$ is linear with respect to parameter k whenever $\alpha = O(k)$

Choping bloom components (1)

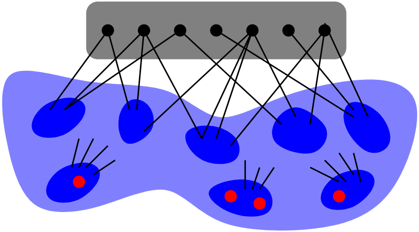
Lemma If $C_1 \dots C_\ell$ is a collection of connected pairwise disjoint subsets of $V(G) \setminus X$ such that $1 \leqslant i \leqslant \ell$, $|N_X(C_i)| \geqslant r$, then

$$\ell \leqslant (1 + \alpha_r).k$$



Lemma If $C_1 \ldots C_\ell$ is a collection of connected pairwise disjoint subsets of $V(G) \setminus X$ such that $1 \leqslant i \leqslant \ell$, $|N_X(C_i)| \geqslant r$, then

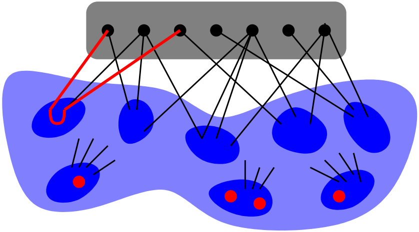
$$\ell \leqslant (1 + \alpha_r).k$$



Choping bloom components (1)

Lemma If $C_1 \ldots C_\ell$ is a collection of connected pairwise disjoint subsets of $V(G) \setminus X$ such that $1 \leqslant i \leqslant \ell$, $|N_X(C_i)| \geqslant r$, then

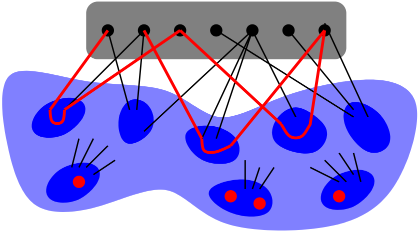
$$\ell \leqslant (1 + \alpha_r).k$$



- Packing subgraphs with large neighborhood in X and edge simulations ideas from [Langer *et al.*'12]

Lemma If $C_1 \dots C_\ell$ is a collection of connected pairwise disjoint subsets of $V(G) \setminus X$ such that $1 \leqslant i \leqslant \ell$, $|N_X(C_i)| \geqslant r$, then

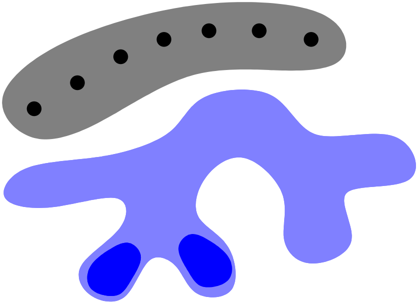
$$\ell \leqslant (1 + \alpha_r).k$$



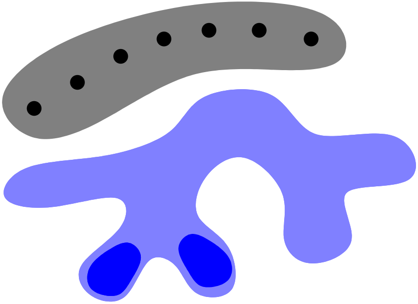
Proposition [Thomason] $\exists$ a constant α_r such that any graph with no K_r-minor has at most $\alpha_r.n = (\alpha.r\sqrt{\log r}).n$ edges.

(we fix $r = |V(H)|$, for H the largest planar graph in $\mathcal{F}$)

Consider a optimal tree-decomposition $\mathcal{T} = (T, \mathcal{B})$ of a "bloom" connected component C of $G - X$ (i.e. $|N_X(C)| \geqslant r$)



Consider a optimal tree-decomposition $\mathcal{T} = (T, \mathcal{B})$ of a "bloom" connected component C of $G - X$ (i.e. $|N_X(C)| \geqslant r$)

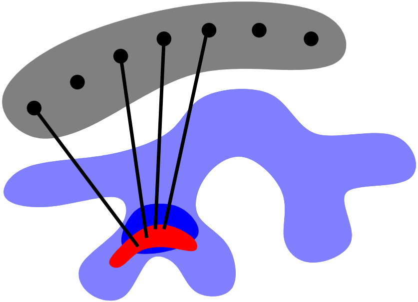


Bottom-up BAG-MARKING algorithm

if B is the LCA of two marked bags of $\mathcal{M}$, or
G_B contains a connected large component, then

- $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove vertices from V_B in bags of $\mathcal{T}$

Consider a optimal tree-decomposition $\mathcal{T} = (T, \mathcal{B})$ of a "bloom" connected component C of $G - X$ (i.e. $|N_X(C)| \geqslant r$)



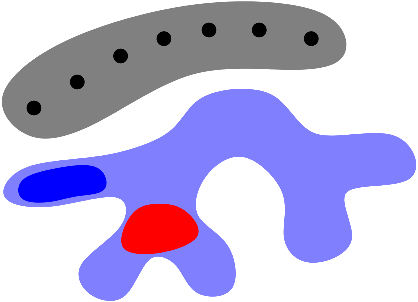
Bottom-up BAG-MARKING algorithm

if B is the LCA of two marked bags of $\mathcal{M}$, or
G_B contains a connected large component, then

- $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove vertices from V_B in bags of $\mathcal{T}$

Choping bloom components (2)

Consider a optimal tree-decomposition $\mathcal{T} = (T, \mathcal{B})$ of a "bloom" connected component C of $G - X$ (i.e. $|N_X(C)| \geqslant r$)



Bottom-up Bag-Marking algorithm

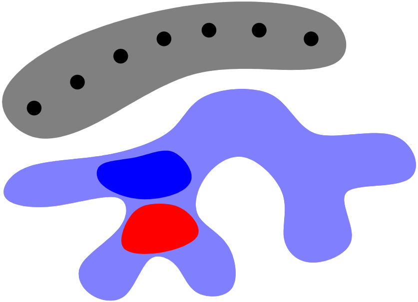
if B is the LCA of two marked bags of $\mathcal{M}$, or

G_B contains a connected large component, then

- $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove vertices from V_B in bags of $\mathcal{T}$

Choping bloom components (2)

Consider a optimal tree-decomposition $\mathcal{T} = (T, \mathcal{B})$ of a "bloom" connected component C of $G - X$ (i.e. $|N_X(C)| \geqslant r$)



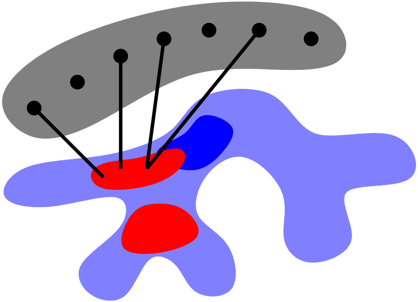
Bottom-up Bag-Marking algorithm

if B is the LCA of two marked bags of $\mathcal{M}$, or
G_B contains a connected large component, then

- $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove vertices from V_B in bags of $\mathcal{T}$

Choping bloom components (2)

Consider a optimal tree-decomposition $\mathcal{T} = (T, \mathcal{B})$ of a "bloom" connected component C of $G - X$ (i.e. $|N_X(C)| \geqslant r$)



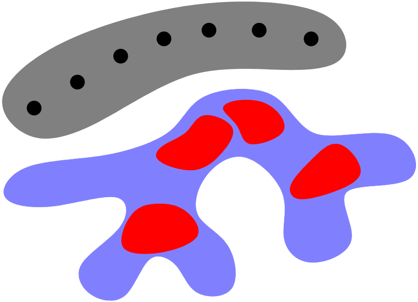
Bottom-up Bag-Marking algorithm

if B is the LCA of two marked bags of $\mathcal{M}$, or
G_B contains a connected large component, then

- $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove vertices from V_B in bags of $\mathcal{T}$

Choping bloom components (2)

Consider a optimal tree-decomposition $\mathcal{T} = (T, \mathcal{B})$ of a "bloom" connected component C of $G - X$ (i.e. $|N_X(C)| \geqslant r$)

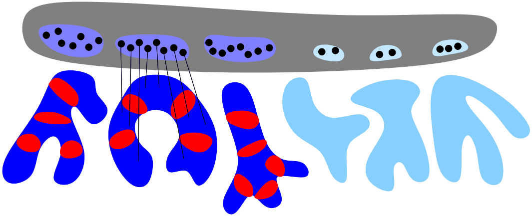


Bottom-up Bag-Marking algorithm

if B is the LCA of two marked bags of $\mathcal{M}$, or
G_B contains a connected large component, then

- $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove vertices from V_B in bags of $\mathcal{T}$

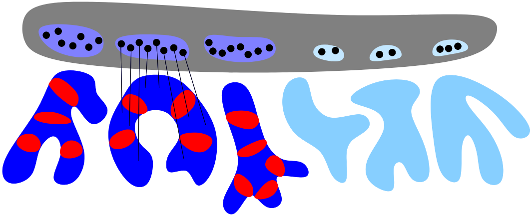
CHOPING BLOOM COMPONENTS (3)



Lemma If (G, X, k) is a YES-instance, then

- $Y_0 = X \cup V(\mathcal{M})$ has size at most $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$

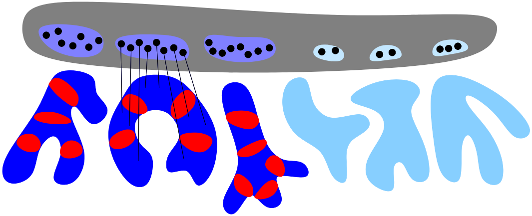
Choping bloom components (3)



Lemma If (G, X, k) is a Yes-instance, then

- $Y_0 = X \cup V(\mathcal{M})$ has size at most $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$

$k = |X|$



Lemma If (G, X, k) is a YES-instance, then

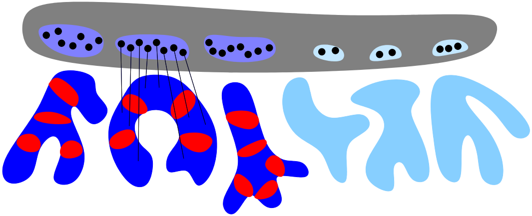
- $Y_0 = X \cup V(\mathcal{M})$ has size at most $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$

$$2t_{\mathcal{F}} = \mathbf{tw}(G - X)$$

Observation: If (G, X, k) is a YES-instance, then $\mathbf{tw}(G - X) \leqslant t_{\mathcal{F}}$

- as $\mathcal{F}$ contains a planar graph and X is a solution

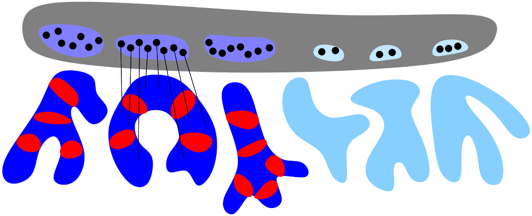
Choping bloom components (3)



Lemma If (G, X, k) is a Yes-instance, then

- $Y_0 = X \cup V(\mathcal{M})$ has size at most $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$

$$|\mathcal{M}| \leqslant (1 + \alpha_r) \cdot k$$



Lemma If (G, X, k) is a YES-instance, then

- $Y_0 = X \cup V(\mathcal{M})$ has size at most $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$
- Every connected component C of $G - Y_0$ satisfies

$$|N_X(C)| \leqslant r \text{ and } |N_{Y_0}(C)| \leqslant r + 2t_{\mathcal{F}}$$

Computing a linear protrusion decomposition

Remark : Y_0 and the connected components of $G - Y_0$ form a protrusion decomposition, but not a linear one!

Branching step 1

- Guess $I = \tilde{X} \cap Y_0$ among the $2^{O(k)}$ subsets of $V(\mathcal{M})$

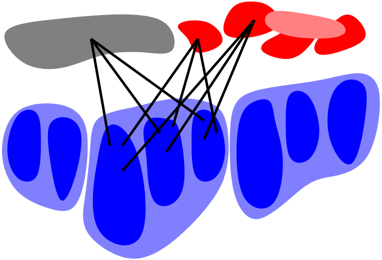
Computing a linear protrusion decomposition

Remark : Y_0 and the connected components of $G - Y_0$ form a protrusion decomposition, but not a linear one!

Branching step 1

- Guess $I = \tilde{X} \cap Y_0$ among the $2^{O(k)}$ subsets of $V(\mathcal{M})$

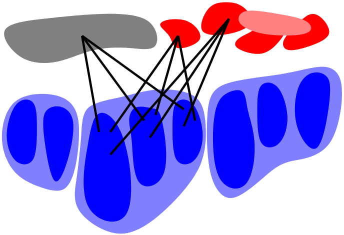
A cluster of $G_I = G - I$ is a maximal set of connected components of $G - Y_0$ with the same neighborhood in $Y_0 \setminus I$



LINEAR PROTRUSION DECOMPOSITION (2)

Lemma If $(G_I, Y_0 \setminus I, k - |I|)$ is a YES-instance of DISJOINT PLANAR-$\mathcal{F}$-DELETION, then the number ℓ of clusters is at most

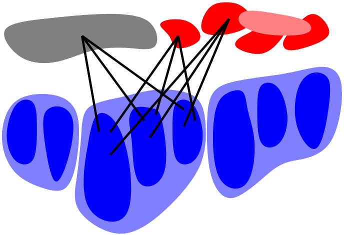
$$(5 t_{\mathcal{F}} \alpha_r \mu_r) \cdot k$$



Linear protrusion decomposition (2)

Lemma If $(G_I, Y_0 \setminus I, k - |I|)$ is a Yes-instance of Disjoint Planar-$\mathcal{F}$-Deletion, then the number ℓ of clusters is at most

$$(5 t_{\mathcal{F}} \alpha_r \mu_r) \cdot k$$

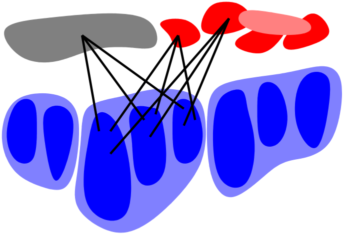


- at most $\ell' = k - |I|$ clusters $C_1 \dots C_{\ell'}$ intersect the alternative solution $\tilde{X}$

Linear protrusion decomposition (2)

Lemma If $(G_I, Y_0 \setminus I, k - |I|)$ is a Yes-instance of Disjoint Planar-$\mathcal{F}$-Deletion, then the number ℓ of clusters is at most

$$(5t_{\mathcal{F}}\alpha_r\mu_r) \cdot k$$

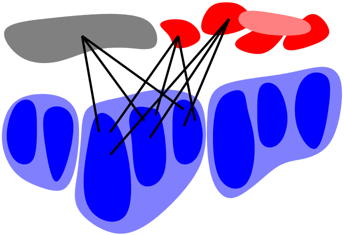


- at most $\ell' = k - |I|$ clusters $C_1 \ldots C_{\ell'}$ intersect the alternative solution $\tilde{X}$
- $G' = G_I - \cup_{i=1}^{\ell'} C_i$ is $\mathcal{F}$-minor free
- using edge simulation we construct a minor of G' on vertices of Y_0

Linear protrusion decomposition (2)

Lemma If $(G_I, Y_0 \setminus I, k - |I|)$ is a Yes-instance of Disjoint Planar-$\mathcal{F}$-Deletion, then the number ℓ of clusters is at most

$$(5 t_{\mathcal{F}} \alpha_r \mu_r) \cdot k$$



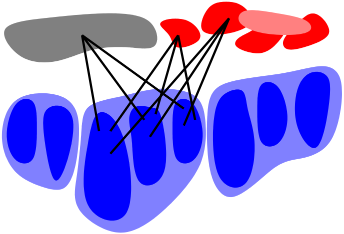
- at most $\ell' = k - |I|$ clusters $C_1 \dots C_{\ell'}$ intersect the alternative solution $\tilde{X}$
- $G' = G_I - \cup_{i=1}^{\ell'} C_i$ is $\mathcal{F}$-minor free
- using edge simulation we construct a minor of G' on vertices of Y_0
- α_r / μ_r correspond to number of edges / cliques in G'

Proposition [FOT] $\exists$ a constant μ st $\forall r > 2$, every n-vertex graph with no K_r-minor has at most $\mu_r.n = 2^{\mu . r \log \log r}.n$ cliques.

Linear protrusion decomposition (2)

Lemma If $(G_I, Y_0 \setminus I, k - |I|)$ is a Yes-instance of Disjoint Planar-$\mathcal{F}$-Deletion, then the number ℓ of clusters is at most

$$(5 t_{\mathcal{F}} \alpha_r \mu_r) \cdot k$$



- The partition $\mathcal{P} = Y_0 \uplus C_1 \uplus \cdots \uplus C_\ell$ is a

$(O(k), r + 2t_{\mathcal{F}})$-protrusion decomposition of $G_I = G - I$

Road map - Recap.

1. Guess a subset I of the alternative solution $\tilde{X}$ such that:
   - $G - I$ has a linear protrusion decomposition
     $$\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$$
   - with $X \subseteq Y_0$ and $\tilde{X} \setminus I \subseteq V \setminus Y_0$
2. Compute $\tilde{X} \setminus I$ using branching on ANNOTATED & COLORED versions of the problem.

Annotated & Colored Disjoint Planar-$\mathcal{F}$-Deletion (1)

- Input : A $(O(k), \beta)$-protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of a graph G,
- Parameter : k
- Question :

Annotated & Colored Disjoint Planar-$\mathcal{F}$-Deletion (1)

- Input : A $(O(k), \beta)$-protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of a graph G, a (k, ℓ)-vector $\overrightarrow{k} = (k, k_0, k_1 \ldots k_\ell)$ and a subset $B \subseteq V \setminus Y_0$
- Parameter : k
- Question :

Annotated & Colored Disjoint Planar-$\mathcal{F}$-Deletion (1)

- Input : A $(O(k), \beta)$-protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of a graph G, a (k, ℓ)-vector $\overrightarrow{k} = (k, k_0, k_1 \ldots k_\ell)$ and a subset $B \subseteq V \setminus Y_0$
- Parameter : k
- Question : Does there exist a set $S \subseteq B \cup Y_0$ such that for every $0 \leqslant i \leqslant \ell$, $|S \cap Y_i| \leqslant k_i$ and $G - S$ is $\mathcal{F}$-minor free.

Denote $(G, \mathcal{P}, \overrightarrow{k}, B)$ an instance of Annotated & Colored Disjoint Planar-$\mathcal{F}$-Deletion

Annotated & Colored Disjoint Planar-$\mathcal{F}$-Deletion (1)

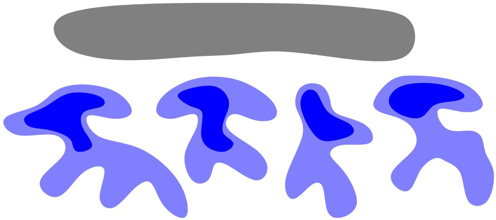
- Input : A $(O(k), \beta)$-protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of a graph G, a (k, ℓ)-vector $\overrightarrow{k} = (k, k_0, k_1 \ldots k_\ell)$ and a subset $B \subseteq V \setminus Y_0$
- Parameter : k
- Question : Does there exist a set $S \subseteq B \cup Y_0$ such that for every $0 \leqslant i \leqslant \ell$, $|S \cap Y_i| \leqslant k_i$ and $G - S$ is $\mathcal{F}$-minor free.

Denote $(G, \mathcal{P}, \overrightarrow{k}, B)$ an instance of Annotated & Colored Disjoint Planar-$\mathcal{F}$-Deletion

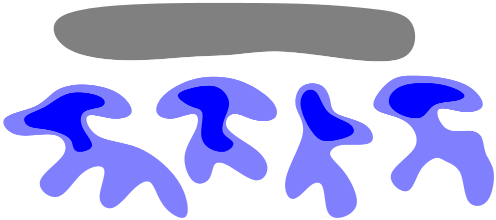
Branching step 2

- Branch on the $2^{O(k)}$ -many possible color vectors
- Initially set B to $V \setminus Y_0$

- Idea : to reduce the set B of annotated vertices, we apply on every part Y_i ($i \neq 0$) a lemma from [Bodlaender *et al.*'09]



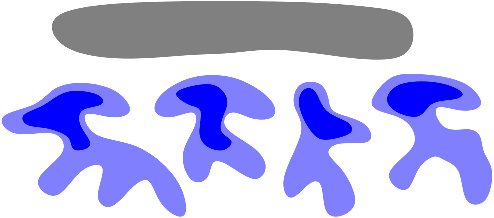
- Idea : to reduce the set B of annotated vertices, we apply on every part Y_i ($i \neq 0$) a lemma from [Bodlaender *et al.*'09]



Lemma $\exists$ a constant $c > 0$ and an $O(n)$-time algorithm that computes a set $B' \subseteq B$ such that

- $|B' \cap Y_i| \leqslant ck_i$ for every $1 \leqslant i \leqslant \ell$

- Idea : to reduce the set B of annotated vertices, we apply on every part Y_i ($i \neq 0$) a lemma from [Bodlaender *et al.*'09]



Lemma $\exists$ a constant $c > 0$ and an $O(n)$-time algorithm that computes a set $B' \subseteq B$ such that

- $|B' \cap Y_i| \leqslant ck_i$ for every $1 \leqslant i \leqslant \ell$
- $(G, \mathcal{P}, \overrightarrow{k}, B)$ is a Yes-instance iff $(G, \mathcal{P}, \overrightarrow{k}, B')$ is a Yes-instance

ANNOTATED & COLORED DISJOINT PLANAR-$\mathcal{F}$-DELETION (3)

Lemma $\exists$ a constant $c > 0$ and a $O(n)$-time algorithm that computes a set $B' \subseteq B$ such that

- $|B' \cap Y_i| \leqslant ck_i$ for every $1 \leqslant i \leqslant \ell$
- $(G, \mathcal{P}, \overrightarrow{k}, B)$ is a YES-inst. iff $(G, \mathcal{P}, \overrightarrow{k}, B')$ is a YES-inst.

Outline of the proof
(we iterate the argument on every Y_i, $B_i = B \cap Y_i$ $(i > 0)$

- Let $Q, Q' \subseteq Y_i$: $Q \equiv_i Q'$ iff $\forall S \in V \setminus Y_i$, $G - (S \cup Q)$ is $\mathcal{F}$-minor free $\Leftrightarrow$ $G - (S \cup Q')$ is $\mathcal{F}$-minor free

Lemma $\exists$ a constant $c > 0$ and a $O(n)$-time algorithm that computes a set $B' \subseteq B$ such that

- $|B' \cap Y_i| \leqslant ck_i$ for every $1 \leqslant i \leqslant \ell$
- $(G, \mathcal{P}, \overrightarrow{k}, B)$ is a YES-inst. iff $(G, \mathcal{P}, \overrightarrow{k}, B')$ is a YES-inst.

Outline of the proof
(we iterate the argument on every Y_i, $B_i = B \cap Y_i$ $(i > 0)$

- Let $Q, Q' \subseteq Y_i$: $Q \equiv_i Q'$ iff $\forall S \in V \setminus Y_i$,
  $G - (S \cup Q)$ is $\mathcal{F}$-minor free $\Leftrightarrow$ $G - (S \cup Q')$ is $\mathcal{F}$-minor free
- The number of equivalence classes is finite
  (depends on the length of MSO formula and treewidth)

Annotated & Colored Disjoint Planar-$\mathcal{F}$-Deletion (3)

Lemma $\exists$ a constant $c > 0$ and a $O(n)$-time algorithm that computes a set $B' \subseteq B$ such that

- $|B' \cap Y_i| \leqslant ck_i$ for every $1 \leqslant i \leqslant \ell$
- $(G, \mathcal{P}, \overrightarrow{k}, B)$ is a Yes-inst. iff $(G, \mathcal{P}, \overrightarrow{k}, B')$ is a Yes-inst.

Outline of the proof

(we iterate the argument on every Y_i, $B_i = B \cap Y_i$ $(i > 0)$)

- Let $Q, Q' \subseteq Y_i$: $Q \equiv_i Q'$ iff $\forall S \in V \setminus Y_i$,
  $G - (S \cup Q)$ is $\mathcal{F}$-minor free $\Leftrightarrow$ $G - (S \cup Q')$ is $\mathcal{F}$-minor free
- The number of equivalence classes is finite
  (depends on the length of MSO formula and treewidth)
- Let Q_i be the union of min-size representants of equiv. classes

ANNOTATED & COLORED DISJOINT PLANAR-$\mathcal{F}$-DELETION (3)

Lemma $\exists$ a constant $c > 0$ and a $O(n)$-time algorithm that computes a set $B' \subseteq B$ such that

- $|B' \cap Y_i| \leqslant ck_i$ for every $1 \leqslant i \leqslant \ell$
- $(G, \mathcal{P}, \overrightarrow{k}, B)$ is a YES-inst. iff $(G, \mathcal{P}, \overrightarrow{k}, B')$ is a YES-inst.

Outline of the proof

(we iterate the argument on every Y_i, $B_i = B \cap Y_i$ $(i > 0)$

- Let $Q, Q' \subseteq Y_i$: $Q \equiv_i Q'$ iff $\forall S \in V \setminus Y_i$, $G - (S \cup Q)$ is $\mathcal{F}$-minor free $\Leftrightarrow$ $G - (S \cup Q')$ is $\mathcal{F}$-minor free
- The number of equivalence classes is finite (depends on the length of MSO formula and treewidth)
- Let Q_i be the union of min-size representants of equiv. classes
- Set $\boxed{B' = (B \setminus B_i) \cup Q_i}$

Recap. of the algorithm

- Reduce Planar-$\mathcal{F}$-Deletion to Disjoint Planar-$\mathcal{F}$-Deletion using iterative compression technic
- Compute a protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus Y_\ell$ with $X \in Y_0$ using the bag marking algorithm
- [Branching step 1] Guess the intersection $I = \tilde{X} \cap (Y_0 \setminus X)$ with $\tilde{X}$ alternative solution disjoint from X

  $\mathcal{P}$ leads to a linear protrusion decomposition of $G - I$
- [Branching step 2] Using a color vector, guess the intersection size k_i of every protrusion Y_i ($i > 0$) with $\tilde{X}$
- Search space reduction step Reduce the set of annotated vertices of size to size $O(k)$
- [Branching step 3] Branch on every possible subsets of the annotated set of vertices

Conclusions

Theorem There exists a $2^{O(k)} \cdot n$ algorithm that solves the Disjoint Planar-$\mathcal{F}$-Deletion problem. And thereby Planar-$\mathcal{F}$-Deletion can be solved in single-exponential time.

Conclusions

Theorem There exists a $2^{O(k)} \cdot n$ algorithm that solves the DISJOINT PLANAR-$\mathcal{F}$-DELETION problem. And thereby PLANAR-$\mathcal{F}$-DELETION can be solved in single-exponential time.

- Instead of a set X such that $G - X$ is $\mathcal{F}$-minor free, it is sufficient that X is a treewidth-bounding set.
- The technique for the ANNOTATED & COLORED version applies to every P-MIN-CMSO problem.
- Using the annotated protrusion reduction rule of [Bodlaender *et al.*'09], we also prove linear kernels on H-topological minor free graphs for any paramatererized problem that has finite index and is treewidth bounding
  - TREEWIDTH-t VERTEX DELETION
  - EDGE DOMINATING SET
  - ...

Thank you. . .