

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)

LaBRI

April 3, 2013

[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 Robertson 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 algorithm
- ▶ [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 algorithm
- ▶ [Fomin et al.'12] $2^{O(k)} \cdot n \log^2 n$ -time algorithm for
CONNECTED-PLANAR- \mathcal{F} -DELETION

OUR RESULT:

A $2^{O(k)} \cdot n^2$ -time algorithm for PLANAR- \mathcal{F} -DELETION

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

ITERATIVE COMPRESSION (1)

Technique used for parameterized minimization problem (Π, k) such as:

- ▶ ODD CYCLE TRANSVERSAL
- ▶ VERTEX COVER
- ▶ FEEDBACK VERTEX SET

ITERATIVE COMPRESSION (1)

Technique used for parameterized minimization problem (Π, k) such as:

- ▶ ODD CYCLE TRANSVERSAL
- ▶ VERTEX COVER
- ▶ FEEDBACK VERTEX SET

1. **Compression step:** Given a solution S of size $k + 1$ for problem Π on the graph G
 - ▶ Either compute an improved solution S' of size at most k
 - ▶ Or decide that such an improved solution does not exist

ITERATIVE COMPRESSION (1)

Technique used for parameterized minimization problem (Π, k) such as:

- ▶ ODD CYCLE TRANSVERSAL
- ▶ VERTEX COVER
- ▶ FEEDBACK VERTEX SET

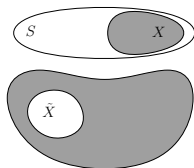
1. **Compression step:** Given a solution S of size $k + 1$ for problem Π on the graph G
 - ▶ Either compute an improved solution S' of size at most k
 - ▶ Or decide that such an improved solution does not exist
2. **Iteration steps:** Iteratively solve the problem Π on instances $(G_i = G[x_1, \dots, x_i], k)$
 - ▶ $S_i = S_{i-1} \cup \{x_i\}$ is a solution of size at most $k + 1$ for G_i
 - ▶ Apply the compression step to G_i and S_i

ITERATIVE COMPRESSION (2)

To solve the **compression step** on graph G with solution S of size $k + 1$

- ▶ Consider every possible (non-empty) subset $X \subset S$, compute the smallest solution \tilde{S} such that $S \setminus X = S \cap \tilde{S}$

Observation: X is a solution for $G - (S \setminus X)$.

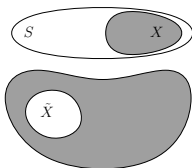


ITERATIVE COMPRESSION (2)

To solve the **compression step** on graph G with solution S of size $k + 1$

- ▶ Consider every possible (non-empty) subset $X \subset S$, compute the smallest solution \tilde{S} such that $S \setminus X = S \cap \tilde{S}$

Observation: X is a solution for $G - (S \setminus X)$.



Or equivalently, solve the **parameterized DISJOINT VERSION** of Π

- ▶ Given an **initial solution** X , compute an **alternative solution** \tilde{X} such that
 - ▶ $X \cap \tilde{X} = \emptyset$ and $|\tilde{X}| < |X|$
 - ▶ parameter is $k = |X|$

ITERATIVE COMPRESSION (3)

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 $|\tilde{X}| \leq |X| = k$?

We call \tilde{X} an **alternative** solution.

ITERATIVE COMPRESSION (3)

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 $|\tilde{X}| \leq |X| = k$?

We call \tilde{X} an **alternative** solution.

Lemma [iterative compression]

If DISJOINT PLANAR- \mathcal{F} -DELETION can be solved in time $O^*(c^k)$ (with $c \in \mathbb{N}^+$), then PLANAR- \mathcal{F} -DELETION can be solved in time $O^*((c+1)^k)$.

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

Observation If (G, X) is a positive instance of DISJOINT PLANAR- \mathcal{F} -DELETION, then

- ▶ $G[X]$ is \mathcal{F} -minor free
- ▶ $G[V \setminus X]$ is \mathcal{F} -minor free

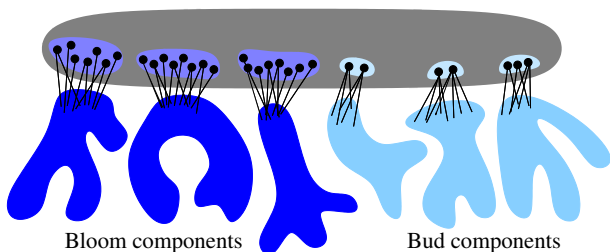
Working Hypothesis : An alternative solution \tilde{X} does exist !

Observation If (G, X) is a positive instance of DISJOINT PLANAR- \mathcal{F} -DELETION, then

- ▶ $G[X]$ is \mathcal{F} -minor free
- ▶ $G[V \setminus X]$ is \mathcal{F} -minor free

Let $r = |H|$ for H being the largest graph in the family \mathcal{F} .

A connected component C of $G - X$ is called a **bloom** component if $|N_X(C)| \geq r$ and a **bud** component otherwise.



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 $\tilde{X} \setminus I \subseteq V \setminus Y_0$

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.

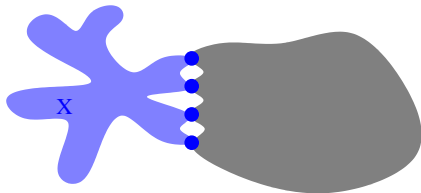
Based on **annotated reduction Lemma** from [Bodlaender *et al.*'09]

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

PROTRUSION DECOMPOSITION (1) - DEFINITION

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

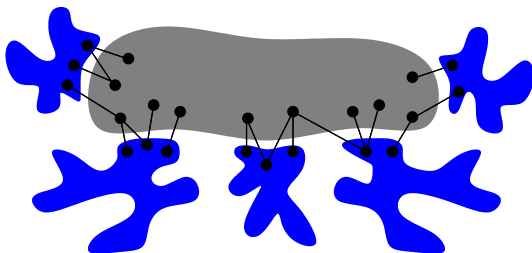
$$|\partial(Y)| \leq \beta \text{ and } \mathbf{tw}(G[Y]) \leq \beta$$



PROTRUSION DECOMPOSITION (1) - DEFINITION

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

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



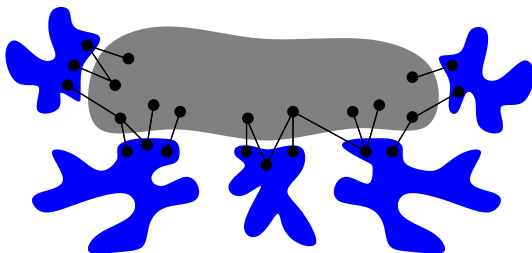
A partition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \dots \uplus Y_\ell$ with $\max\{\ell, |Y_0|\} \leq \alpha$ is an (α, β) -**protrusion decomposition** if for every $1 \leq i \leq \ell$,

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

PROTRUSION DECOMPOSITION (1) - DEFINITION

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

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



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

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

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

PROTRUSION DECOMPOSITION (2)

- ▶ **Protrusion replacement techniques** have been used in meta-algorithmic results in the context of **kernelization**:
 - ▶ Region decomposition on planar graphs [Guo and Niedermeier'07]
 - ▶ Protrusion decomposition on bounded genus graphs [Bodlaender et al'09]
 - ▶ Generalization to H -minor free graphs [Fomin et al'10]

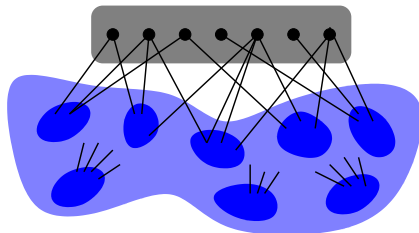
PROTRUSION DECOMPOSITION (2)

- ▶ **Protrusion replacement techniques** have been used in meta-algorithmic results in the context of **kernelization**:
 - ▶ Region decomposition on planar graphs [Guo and Niedermeier'07]
 - ▶ Protrusion decomposition on bounded genus graphs [Bodlaender et al'09]
 - ▶ Generalization to H -minor free graphs [Fomin et al'10]
- ▶ The **ideas** from protrusion decomposition replacement techniques come from **dynamic programming** techniques on graphs of bounded treewidth developed in
 - ▶ Courcelle's theorem
 - ▶ de Fluter and Bodlaender treewidth reduction algorithms

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 $1 \leq i \leq \ell$, $|N_X(C_i)| \geq r$, then

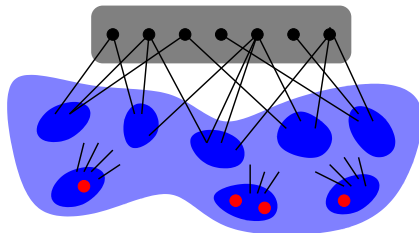
$$\ell \leq (1 + \alpha_r) \cdot k$$



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 $1 \leq i \leq \ell$, $|N_X(C_i)| \geq r$, then

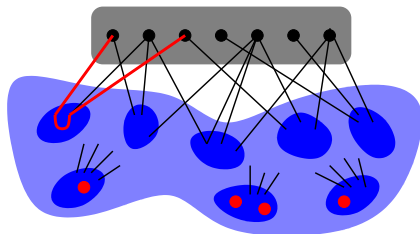
$$\ell \leq (1 + \alpha_r) \cdot k$$



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 $1 \leq i \leq \ell$, $|N_X(C_i)| \geq r$, then

$$\ell \leq (1 + \alpha_r) \cdot k$$

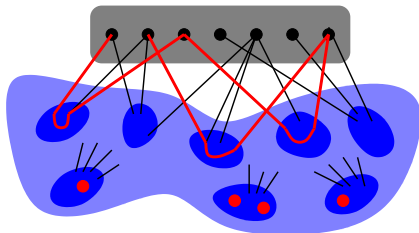


- ▶ Packing subgraphs with large neighborhood in X and edge simulations ideas from [Langer *et al.*'12]

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 $1 \leq i \leq \ell$, $|N_X(C_i)| \geq r$, then

$$\ell \leq (1 + \alpha_r) \cdot k$$

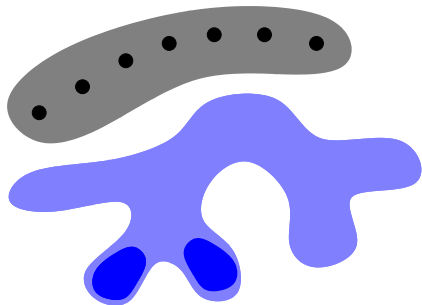


Proposition [Thomason] \exists a constant α_r such that any graph with no K_r -minor has at most $\alpha_r \cdot n = (\alpha_r \cdot r \sqrt{\log r}) \cdot n$ edges.

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

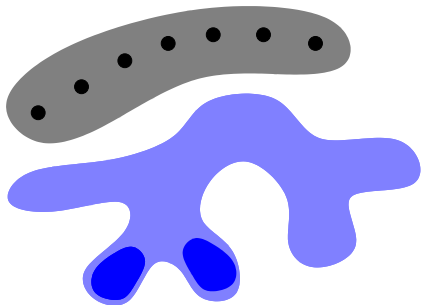
CHOPING BLOOM COMPONENTS (2)

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



CHOPING BLOOM COMPONENTS (2)

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



Bottom-up BAG-MARKING algorithm

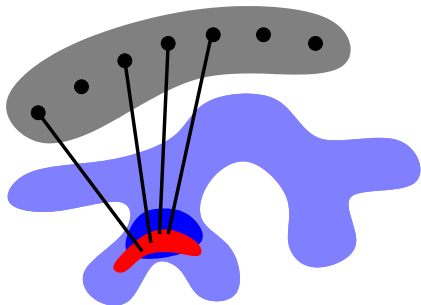
if B is the **LCA** of two marked bags of \mathcal{M} , or

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



Bottom-up BAG-MARKING algorithm

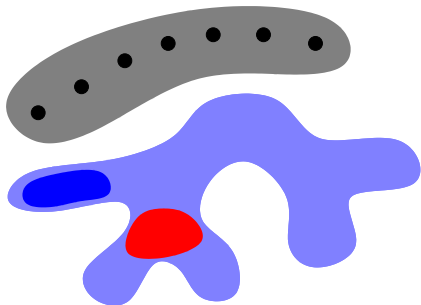
if B is the **LCA** of two marked bags of \mathcal{M} , or

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



Bottom-up BAG-MARKING algorithm

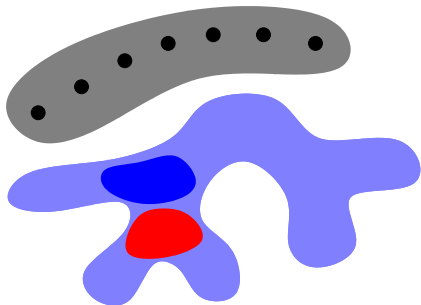
if B is the **LCA** of two marked bags of \mathcal{M} , or

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



Bottom-up BAG-MARKING algorithm

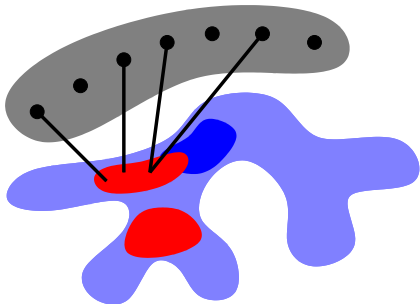
if B is the **LCA** of two marked bags of \mathcal{M} , or

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



Bottom-up BAG-MARKING algorithm

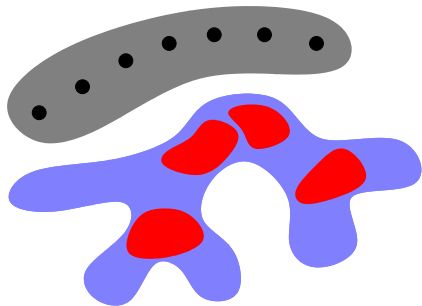
if B is the **LCA** of two marked bags of \mathcal{M} , or

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



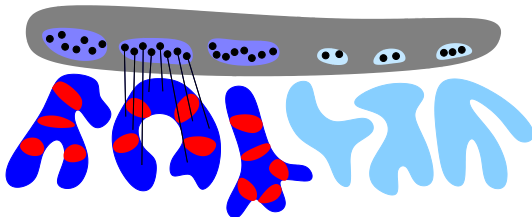
Bottom-up BAG-MARKING algorithm

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}

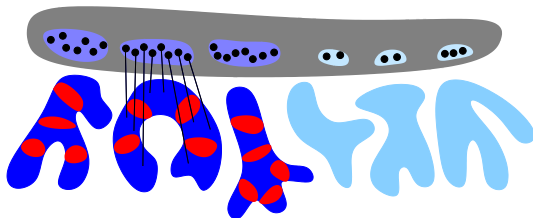
CHOPING BLOOM COMPONENTS (3)



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

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

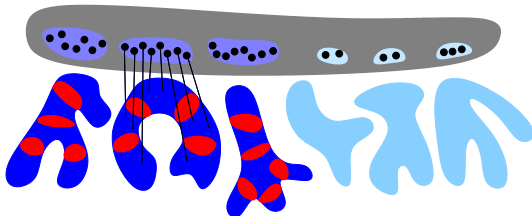
CHOPING BLOOM COMPONENTS (3)



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

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

CHOPING BLOOM COMPONENTS (3)



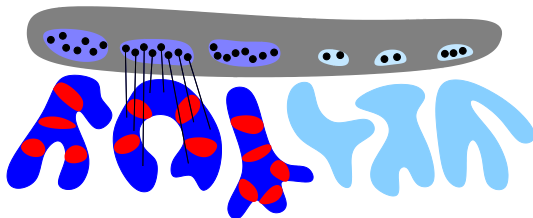
Lemma If (G, X, k) is a YES-instance, then

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

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

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

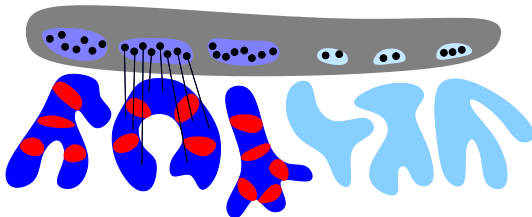
CHOPING BLOOM COMPONENTS (3)



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

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

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 $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$
- ▶ Every connected component C of $G - Y_0$ satisfies
 $|N_X(C)| \leq r$ and $|N_{Y_0}(C)| \leq r + 2t_{\mathcal{F}}$

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!

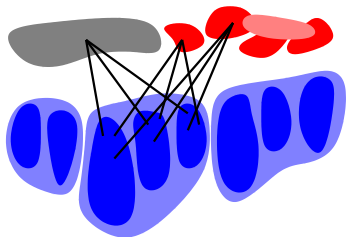
BRANCHING STEP 1

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

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

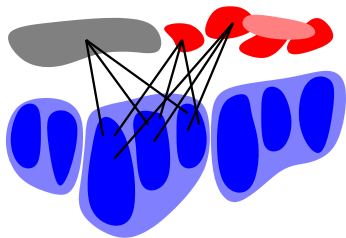
$$(5t_{\mathcal{F}}\alpha_r\mu_r) \cdot k$$



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

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

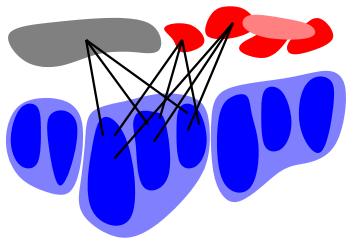


- ▶ at most $\ell' = k - |I|$ clusters $C_1 \dots C_{\ell'}$ intersect the alternative solution \tilde{X}

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

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

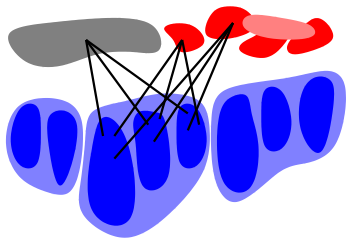


- ▶ at most $\ell' = k - |I|$ clusters $C_1 \dots C_{\ell'}$ intersect the alternative solution \tilde{X}
- ▶ $G' = G_I - \cup_{i=1}^{\ell'} C_i$ is \mathcal{F} -minor free
- ▶ using edge simulation we construct a minor of G' on vertices of Y_0

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

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



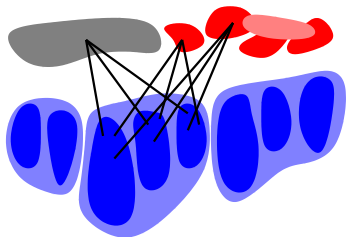
- ▶ at most $\ell' = k - |I|$ clusters $C_1 \dots C_{\ell'}$ intersect the alternative solution \tilde{X}
- ▶ $G' = G_I - \cup_{i=1}^{\ell'} C_i$ is \mathcal{F} -minor free
- ▶ using edge simulation we construct a minor of G' on vertices of Y_0
- ▶ α_r / μ_r correspond to number of edges / cliques in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

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

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



- ▶ The partition $\mathcal{P} = Y_0 \uplus C_1 \uplus \dots \uplus C_\ell$ 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 $\mathcal{P} = Y_0 \uplus Y_1 \uplus \dots \uplus Y_\ell$ of a graph G , a (k, ℓ) -vector $\vec{k} = (k, k_0, k_1 \dots k_\ell)$ 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 $\mathcal{P} = Y_0 \uplus Y_1 \uplus \dots \uplus Y_\ell$ of a graph G , a (k, ℓ) -vector $\vec{k} = (k, k_0, k_1 \dots k_\ell)$ 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 $0 \leq i \leq \ell$, $|S \cap Y_i| \leq k_i$ and $G - S$ is \mathcal{F} -minor free.

Denote $(G, \mathcal{P}, \vec{k}, B)$ 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 $\mathcal{P} = Y_0 \uplus Y_1 \uplus \dots \uplus Y_\ell$ of a graph G , a (k, ℓ) -vector $\vec{k} = (k, k_0, k_1 \dots k_\ell)$ 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 $0 \leq i \leq \ell$, $|S \cap Y_i| \leq k_i$ and $G - S$ is \mathcal{F} -minor free.

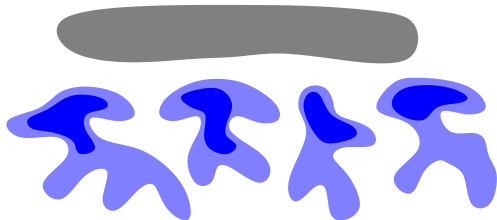
Denote $(G, \mathcal{P}, \vec{k}, B)$ an instance of ANNOTATED & COLORED DISJOINT PLANAR- \mathcal{F} -DELETION

BRANCHING STEP 2

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

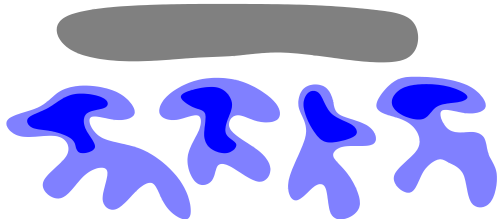
ANNOTATED & COLORED DISJOINT PLANAR- \mathcal{F} -DELETION (2)

- ▶ **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]



ANNOTATED & COLORED DISJOINT PLANAR- \mathcal{F} -DELETION (2)

- ▶ **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]

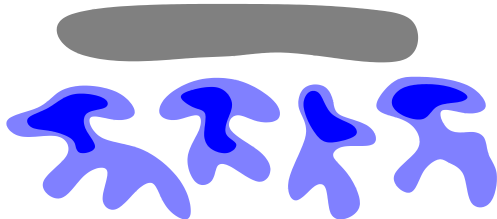


Lemma \exists a constant $c > 0$ and an $O(n)$ -time algorithm that computes a set $B' \subseteq B$ such that

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

ANNOTATED & COLORED DISJOINT PLANAR- \mathcal{F} -DELETION (2)

- ▶ **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

- ▶ $|B' \cap Y_i| \leq ck_i$ for every $1 \leq i \leq \ell$
- ▶ $(G, \mathcal{P}, \vec{k}, B)$ is a YES-instance iff $(G, \mathcal{P}, \vec{k}, B')$ 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

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

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

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

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

- ▶ $|B' \cap Y_i| \leq ck_i$ for every $1 \leq i \leq \ell$
- ▶ $(G, \mathcal{P}, \vec{k}, B)$ is a YES-inst. iff $(G, \mathcal{P}, \vec{k}, B')$ 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 $B' = (B \setminus B_i) \cup Q_i$

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 parameterized problem that has finite index and is treewidth bounding
 - ▶ TREEWIDTH- t VERTEX DELETION
 - ▶ EDGE DOMINATING SET
 - ▶ ...

Thank you...