A single exponential FPT algorithm for the K_4-minor cover problem

Eunjung Kim

CNRS - LAMSADE, Paris, France

Joint work with
Christophe Paul (CNRS - LIRMM, FRANCE
Geevarghese Philip (MPI, GERMANY)

July 4, 2012

Parameterized K_4-Minor Cover

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

- at most k vertices $S \subseteq V$ s.t $G[V \setminus S]$ is K_4-minor free ?

Parameterized K_4-Minor Cover
(a.k.a. Parameterized Treewidth-two Vertex Deletion)

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

- at most k vertices $S \subseteq V$ s.t $G[V \setminus S]$ is K_4-minor free ?
- at most k vertices $S \subseteq V$ s.t $tw(G[V \setminus S]) \leqslant 2$?

Parameterized K_4-Minor Cover
(a.k.a. Parameterized Treewidth-two Vertex Deletion)

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

- at most k vertices $S \subseteq V$ s.t $G[V \setminus S]$ is K_4-minor free ?
- at most k vertices $S \subseteq V$ s.t $tw(G[V \setminus S]) \leqslant 2$?

Observations :

1. Vertex Cover $\equiv$ K_2-Minor Cover
   $\equiv$ Treewidth-zero Vertex Deletion
2. Feedback Vertex Set
   $\equiv$ K_3-Minor Cover
   $\equiv$ Treewidth-one Vertex Deletion

More generally,

How fast can we solve Treewidth-t Vertex Deletion ?

Known results (*when we submitted)

1. Parameterized K_4-Minor Cover is FPT (by the Roberston and Seymour' graph minor theorem or by Courcelle's theorem)
2. Best algorithm runs in $2^{O(k \log k)} \cdot n^{O(1)}$ [Fomin et al.'11]
3. $2^{O(k)} \cdot n^{O(1)}$-algorithm when for $t = 0, 1$.
4. No hope for a $2^{o(k)}.n^{O(1)}$ algorithm [Chen et al.'05]

Known results (*when we submitted)

1. Parameterized K_4-Minor Cover is FPT (by the Roberston and Seymour' graph minor theorem or by Courcelle's theorem)
2. Best algorithm runs in $2^{O(k \log k)} \cdot n^{O(1)}$ [Fomin et al.'11]
3. $2^{O(k)} \cdot n^{O(1)}$-algorithm when for $t = 0, 1$.
4. No hope for a $2^{o(k)}.n^{O(1)}$ algorithm [Chen et al.'05]

Our result

There exists an algorithm that solves the Parameterized K_4-Minor Cover problem in time $2^{O(k)} \cdot n^{O(1)}$.

Known results (*when we submitted)

1. Parameterized K_4-Minor Cover is FPT
   (by the Roberston and Seymour' graph minor theorem or by Courcelle's theorem)
2. Best algorithm runs in $2^{O(k \log k)} \cdot n^{O(1)}$ [Fomin et al.'11]
3. $2^{O(k)} \cdot n^{O(1)}$-algorithm when for $t = 0, 1$.
4. No hope for a $2^{o(k)}.n^{O(1)}$ algorithm [Chen et al.'05]

Our result

There exists an algorithm that solves the Parameterized K_4-Minor Cover problem in time $2^{O(k)} \cdot n^{O(1)}$.

Known results (*now, a few months later...)

1. Treewidth-t Vertex Deletion in $2^{O(k)} \cdot n \log n^2$ [Fomin et al.'12], in $2^{O(k)} \cdot n^2$ [Kim et al.'12]
2. Polynomial kernel [Fomin et al.'12]

Iterative compression

allows us to focus on DISJOINT-K_4-MINOR COVER

- Given $G = (V, E)$, a K_4-MINOR COVER S of size $k + 1$
- Compute (if it exists) a K_4-MINOR COVER S' of size k such that $S \cap S' = \emptyset$

Iterative compression

allows us to focus on DISJOINT-K_4-MINOR COVER

- Given $G = (V, E)$, a K_4-MINOR COVER S of size $k + 1$
- Compute (if it exists) a K_4-MINOR COVER S' of size k such that $S \cap S' = \emptyset$

Folklore: If DISJOINT-K_4-MINOR COVER is single-exponential, PARAMETERIZED K_4-MINOR COVER is single-exponential.

Iterative compression

allows us to focus on DISJOINT-K_4-MINOR COVER

- Given $G = (V, E)$, a K_4-MINOR COVER S of size $k + 1$
- Compute (if it exists) a K_4-MINOR COVER S' of size k such that $S \cap S' = \emptyset$

Folklore: If DISJOINT-K_4-MINOR COVER is single-exponential, PARAMETERIZED K_4-MINOR COVER is single-exponential.

From the additional S, we can retrieve rich structural information.

Iterative compression

allows us to focus on DISJOINT-K_4-MINOR COVER

- Given $G = (V, E)$, a K_4-MINOR COVER S of size $k + 1$
- Compute (if it exists) a K_4-MINOR COVER S' of size k such that $S \cap S' = \emptyset$

Folklore: If DISJOINT-K_4-MINOR COVER is single-exponential, PARAMETERIZED K_4-MINOR COVER is single-exponential.

From the additional S, we can retrieve rich structural information.

Our algorithm for DISJOINT-K_4-MINOR COVER can be viewed as a generalization of [Chen et al.08] for DISJOINT-FVS.

DISJOINT-FEEDBACK VERTEX SET (DISJOINT-FVS)

- Given $G = (V, E)$, a feedback vertex set S of size $k + 1$
- Compute (if it exists) a feedback vertex set S' size k such that $S \cap S' = \emptyset$.

Disjoint-Feedback Vertex Set (Disjoint-FVS)

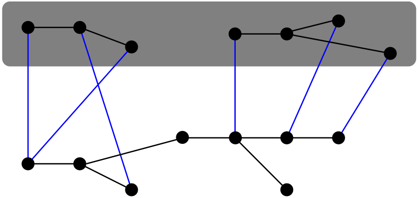
- Given $G = (V, E)$, a feedback vertex set S of size $k + 1$
- Compute (if it exists) a feedback vertex set S' size k such that $S \cap S' = \emptyset$.

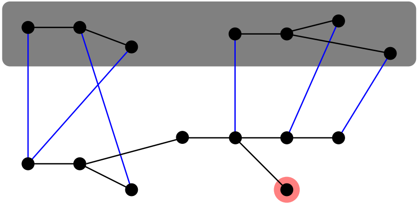
[Chen et al.08] We use

- branching and reduction rules
- a measure function to analyze the time complexity

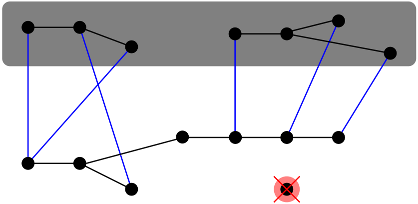
$$\boxed{\mu = k + \#cc(G[S])}$$

Skip example

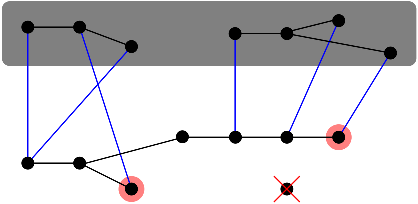




Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

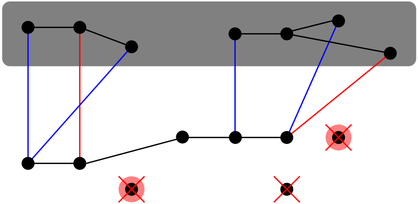


Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$



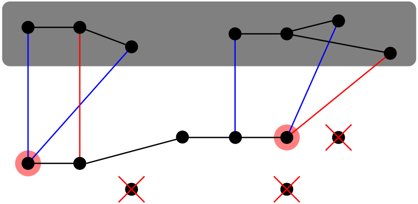
Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$



Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

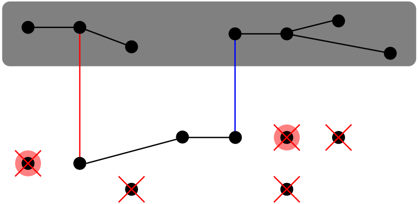
Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$



Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$

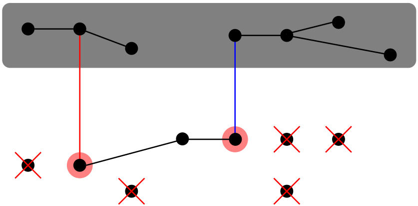
Red. Rule 3: Remove every vertex $x \in V \setminus S$ with at least 2 neighbours in some connect. comp. C of $G[S]$ and decrease k by 1



Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$

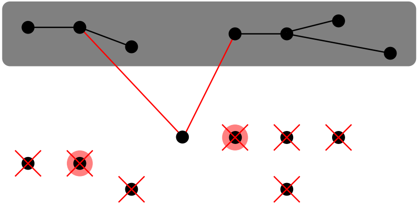
Red. Rule 3: Remove every vertex $x \in V \setminus S$ with at least 2 neighbours in some connect. comp. C of $G[S]$ and decrease k by 1



Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$

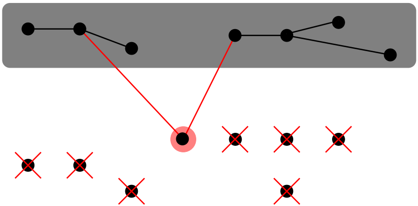
Red. Rule 3: Remove every vertex $x \in V \setminus S$ with at least 2 neighbours in some connect. comp. C of $G[S]$ and decrease k by 1



Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$

Red. Rule 3: Remove every vertex $x \in V \setminus S$ with at least 2 neighbours in some connect. comp. C of $G[S]$ and decrease k by 1



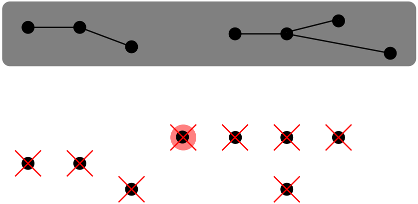
Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$

Red. Rule 3: Remove every vertex $x \in V \setminus S$ with at least 2 neighbours in some connect. comp. C of $G[S]$ and decrease k by 1

Branching Rule: If $x \in V \setminus S$ has two neighbours in two different connected components of $G[S]$, then branch on

- $(G - \{x\}, S, k - 1)$ $\Rightarrow$ μ decreases



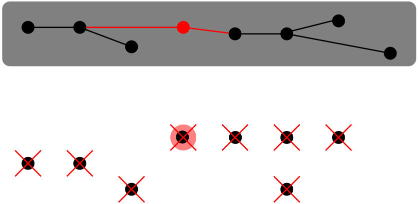
Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$

Red. Rule 3: Remove every vertex $x \in V \setminus S$ with at least 2 neighbours in some connect. comp. C of $G[S]$ and decrease k by 1

Branching Rule: If $x \in V \setminus S$ has two neighbours in two different connected components of $G[S]$, then branch on

- $(G - \{x\}, S, k - 1)$ $\Rightarrow$ μ decreases



Red. Rule 1: Remove leaf $x \in V \setminus S$ if $N(x) \cap S = \emptyset$

Red. Rule 2: Bypass leaf $x \in V \setminus S$ if $d(x) = 2$, $|N(x) \cap S| = 1$

Red. Rule 3: Remove every vertex $x \in V \setminus S$ with at least 2 neighbours in some connect. comp. C of $G[S]$ and decrease k by 1

Branching Rule: If $x \in V \setminus S$ has two neighbours in two different connected components of $G[S]$, then branch on

- $(G - \{x\}, S, k - 1)$ $\Rightarrow \mu$ decreases
- $(G, S \cup \{x\}, k)$ $\Rightarrow \mu$ decreases

Ingredients of [Chen et al.08]

- Branching rules AND appropriate measure function μ.
- Reduction rules to bound the branching degree.
- An appropriate tree-like structure to process $G - S$.
- In the search tree, leaf instances are not hard.

Ingredients of [Chen et al.08]

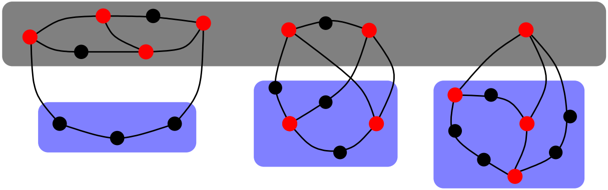
- Branching rules AND appropriate measure function μ.
- Reduction rules to bound the branching degree.
- An appropriate tree-like structure to process $G - S$.
- In the search tree, leaf instances are not hard.

For the Disjoint-K_4-Minor Cover we have

- adapted the branching rules and introduce a new measure μ
- adapted the reduction rules (extended bypassing + chandelier + trivial)
- extended SP-decomposition for treewidth-2 graphs.
- in the search tree, a leaf instance is Vertex Cover on circle graphs (polytime).

Branching Rules (1)

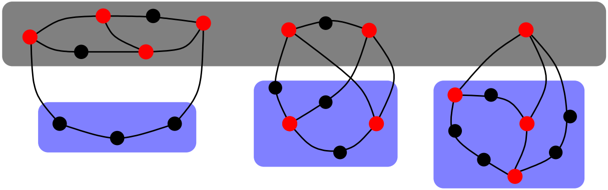
Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover.



Branching Rule 1: If $X \subseteq V \setminus S$ is a set such that $G[S \cup X]$ contains a K_4-subdivision, then

Branching Rules (1)

Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover.



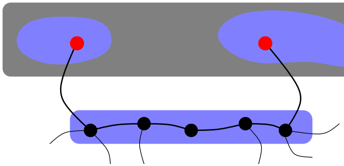
Branching Rule 1: If $X \subseteq V \setminus S$ is a set such that $G[S \cup X]$ contains a K_4-subdivision, then

- we must delete one of X.

Branching Rules (2)

Branching Rules (2)

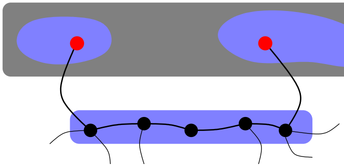
Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover.



Branching Rule 2: If $X \subseteq V \setminus S$ is an s_1, s_2-path and $\{s_1, s_2\} \subseteq N_S(X)$ with $cc_S(s_1) \neq cc_S(s_2)$, then,

Branching Rules (2)

Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover.

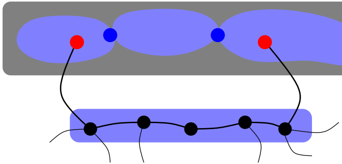


Branching Rule 2: If $X \subseteq V \setminus S$ is an s_1, s_2-path and $\{s_1, s_2\} \subseteq N_S(X)$ with $cc_S(s_1) \neq cc_S(s_2)$, then,

- either we delete one of X
- or X is added to S (no vertex deleted)

Branching Rules (3)

Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover.

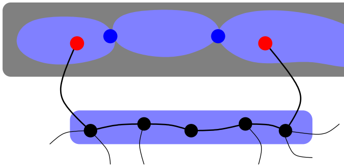


Branching Rule 3: If $X \subseteq V \setminus S$ is an s_1, s_2-path and $\{s_1, s_2\} \subseteq N_S(X)$ with $cc_S(s_1) = cc_S(s_2)$ and $bc_S(s_1) \neq bc_S(s_2)$, then,

- either we delete one of X
- or X is added to S (no vertex deleted)

Branching Rules (3)

Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover.



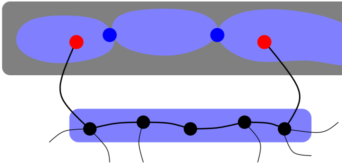
Branching Rule 3: If $X \subseteq V \setminus S$ is an s_1, s_2-path and $\{s_1, s_2\} \subseteq N_S(X)$ with $cc_S(s_1) = cc_S(s_2)$ and $bc_S(s_1) \neq bc_S(s_2)$, then,

- either we delete one of X
- or X is added to S (no vertex deleted)

Claim: $\mu = c_1 \times k + c_1 \times \#cc(G[S]) + \#bc(G[S])$ is decreasing

Branching Rules (3)

Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover.



Branching Rule 3: If $X \subseteq V \setminus S$ is an s_1, s_2-path and $\{s_1, s_2\} \subseteq N_S(X)$ with $cc_S(s_1) = cc_S(s_2)$ and $bc_S(s_1) \neq bc_S(s_2)$, then,

- either we delete one of X
- or X is added to S (no vertex deleted)

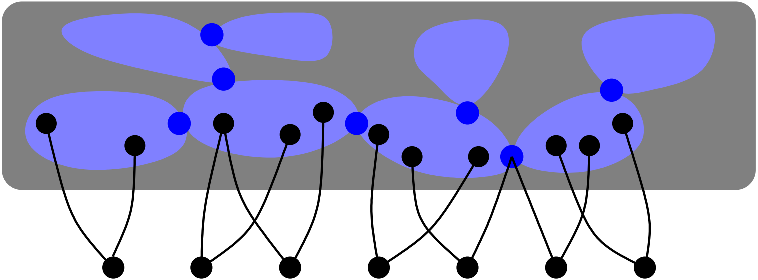
Claim: $\mu = c_1 \times k + c_1 \times \#cc(G[S]) + \#bc(G[S])$ is decreasing

- c_1 depends on the maximum size of the sets X on which the branching rules is applied

Suppose the branching rules have been exhaustively applied to all connected component (no matter how large $|X|$ might be).

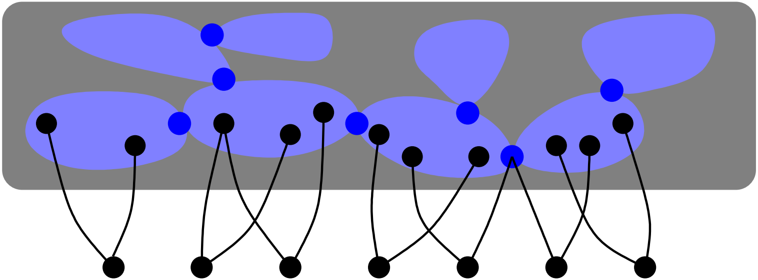
Suppose the branching rules have been exhaustively applied to all connected component (no matter how large $|X|$ might be).

Then the current instance has a nice structure.



Suppose the branching rules have been exhaustively applied to all connected component (no matter how large $|X|$ might be).

Then the current instance has a nice structure.



independent instance can be solved in poly-time

How to bound the size of X to branch on?
First, we develop a notion of tree-like structure for K_4-minor-free graphs.

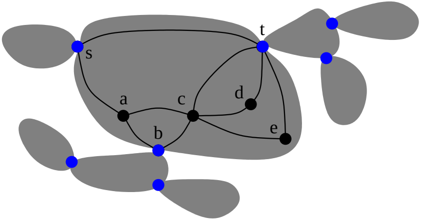
How to bound the size of X to branch on?
First, we develop a notion of tree-like structure for K_4-minor-free graphs.

Use the fact: A graph is K_4-minor free iff its biconnected components are series-parallel graphs

How to bound the size of X to branch on?
First, we develop a notion of tree-like structure for K_4-minor-free graphs.

Use the fact: A graph is K_4-minor free iff its biconnected components are series-parallel graphs

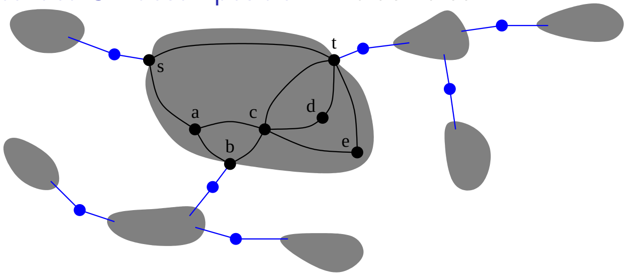
extended-SP decomposition



How to bound the size of X to branch on?
First, we develop a notion of tree-like structure for K_4-minor-free graphs.

Use the fact: A graph is K_4-minor free iff its biconnected components are series-parallel graphs

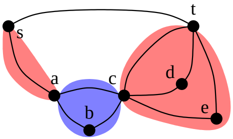
extended-SP decomposition = block tree

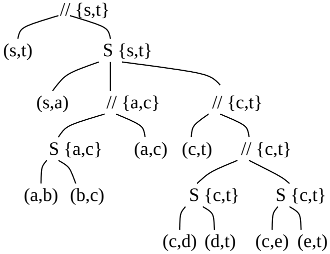


How to bound the size of X to branch on?
First, we develop a notion of tree-like structure for K_4-minor-free graphs.

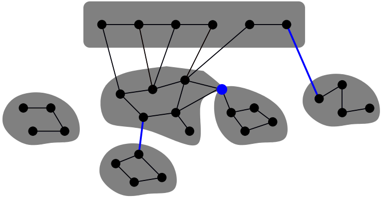
Use the fact: A graph is K_4-minor free iff its biconnected components are series-parallel graphs

extended-SP decomposition = block tree + SP-tree on every block



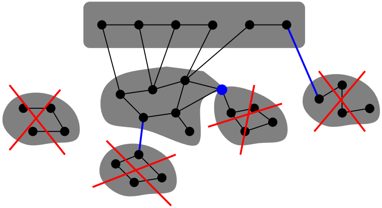


Reduction Rules: trivial



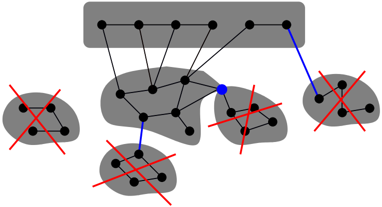
Reduction rule: Components NOT participating any K_4-subdivision is removed.

Reduction Rules: trivial



Reduction rule: Components NOT participating any K_4-subdivision is removed.

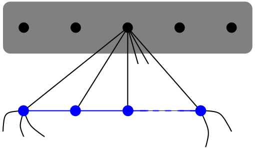
Reduction Rules: trivial



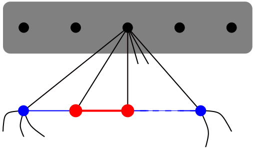
Reduction rule: Components NOT participating any K_4-subdivision is removed.

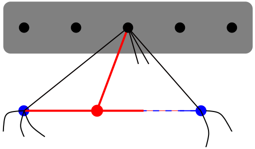
Reduction rule: Bypass degree-2 vertices and remove multiple edges.

Reduction Rules: Chandelier

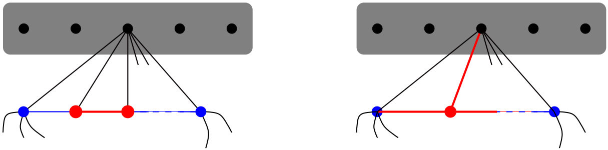


Reduction Rules: Chandelier



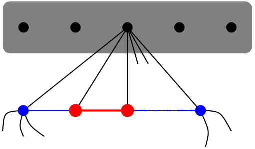


Reduction Rules: Chandelier



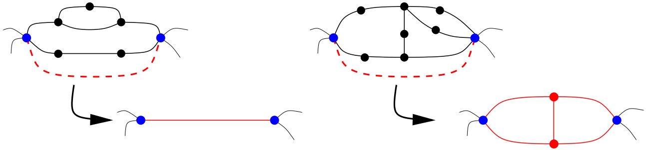
Reduction Rules: extended bypass-1

Reduction Rules: Chandelier



Reduction Rules: extended bypass-1

when connected X s.t. $X \cap S = \emptyset$ has a separator of size 2

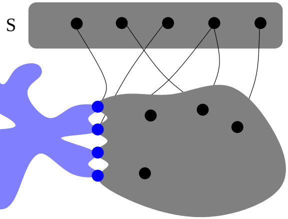


Reduction Rules: extended bypass-2

Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover

Disjoint Protrusion Rule: Let X be a t-protrusion of G such that

$$X \cap S = \emptyset \text{ and } |X| > \gamma(t)$$

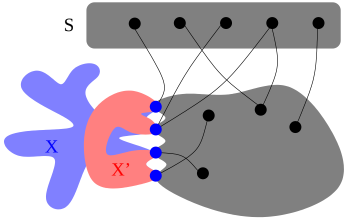


Then,

Let (G, S, k) be an instance of Disjoint-K_4-Minor Cover

Disjoint Protrusion Rule: Let X be a t-protrusion of G such that

$$X \cap S = \emptyset \text{ and } |X| > \gamma(t)$$



Then, replace X with a t-protrusion X' of smaller size.

Algorithm outline

1. Apply reduction rules and branching rules on every set $\{x\}$ (with $x \in V \setminus S$)

1. Apply reduction rules and branching rules on every set $\{x\}$ (with $x \in V \setminus S$)

   Lemma: $|N(v) \cap S| \leq 2$ for all $v \in V - S$.

Algorithm outline

1. Apply reduction rules and branching rules on every set $\{x\}$ (with $x \in V \setminus S$)

   Lemma: $|N(v) \cap S| \leq 2$ for all $v \in V - S$.

2. Use the extended-SP decomposition to apply the branching rules and reduction rules in a bottom-up manner.

Algorithm outline

1. Apply reduction rules and branching rules on every set $\{x\}$ (with $x \in V \setminus S$)

   Lemma: $|N(v) \cap S| \leq 2$ for all $v \in V - S$.

2. Use the extended-SP decomposition to apply the branching rules and reduction rules in a bottom-up manner.

   Branch-or-Reduce Lemma: Either one of the branching rules apply on $|X| \leq c_1$, or extended bypassing rules apply. Otherwise you're at a leaf instance.

Algorithm outline

1. Apply reduction rules and branching rules on every set $\{x\}$ (with $x \in V \setminus S$)

   Lemma: $|N(v) \cap S| \leq 2$ for all $v \in V - S$.

2. Use the extended-SP decomposition to apply the branching rules and reduction rules in a bottom-up manner.

   Branch-or-Reduce Lemma: Either one of the branching rules apply on $|X| \leq c_1$, or extended bypassing rules apply. Otherwise you're at a leaf instance.

3. Solve each independent instance in polytime

Conclusion

Theorem: There exists a single-exponential FPT-time algorithm for the K_4-Minor Cover problem.

Conclusion

Theorem: There exists a single-exponential FPT-time algorithm for the K_4-Minor Cover problem.

Open question:

1. Due to recent developement, we have $2^{O(k)} \cdot n^{O(1)}$-time algorithm for $\mathcal{F}$-minor cover problem, for any finite collection $\mathcal{F}$ containing at least one planar graph.

Conclusion

Theorem: There exists a single-exponential FPT-time algorithm for the K_4-Minor Cover problem.

Open question:

1. Due to recent developement, we have $2^{O(k)} \cdot n^{O(1)}$-time algorithm for $\mathcal{F}$-minor cover problem, for any finite collection $\mathcal{F}$ containing at least one planar graph.

2. How about $\mathcal{F} = \{K_5\}$ or $\mathcal{F} = \{K_5, K_{3,3}\}$? Current best is double-exponential i.e. $2^{2^{O(k)}}$.

CONCLUSION

Theorem: There exists a single-exponential FPT-time algorithm for the K_4-MINOR COVER problem.

Open question:

1. Due to recent developement, we have $2^{O(k)} \cdot n^{O(1)}$-time algorithm for $\mathcal{F}$-minor cover problem, for any finite collection $\mathcal{F}$ containing at least one planar graph.

2. How about $\mathcal{F} = \{K_5\}$ or $\mathcal{F} = \{K_5, K_{3,3}\}$? Current best is double-exponential i.e. $2^{2^{O(k)}}$.

Thank you