Feedback Vertex Set for pseudo-disk graphs in subexponential FPT time

Gaétan Berthe, LIRMM, Montpellier, France

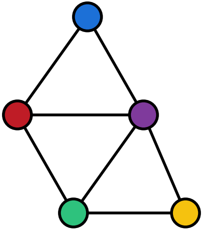
Joint work with

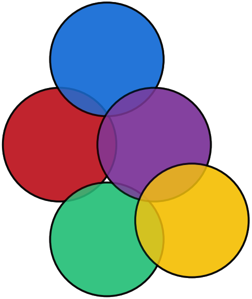
Marin Bougeret, Daniel Gonçalves, Jean-Florent Raymond

WG 2024, 19 June 2024

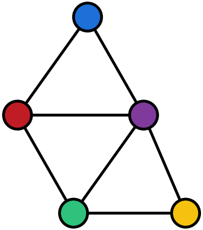
Table of Contents

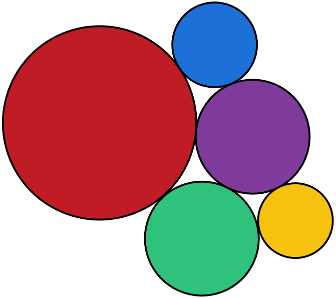
Geometric graphs : Unit disks



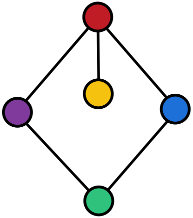


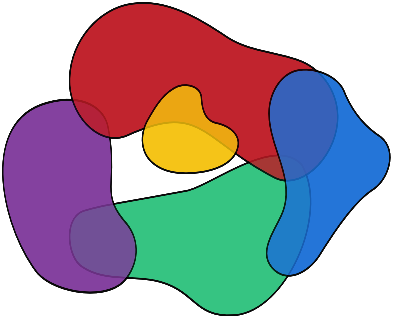
Geometric graphs : Disks in contact = planar graphs



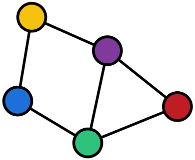


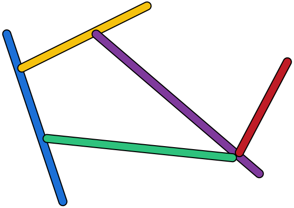
Geometric graphs : Pseudo-disks



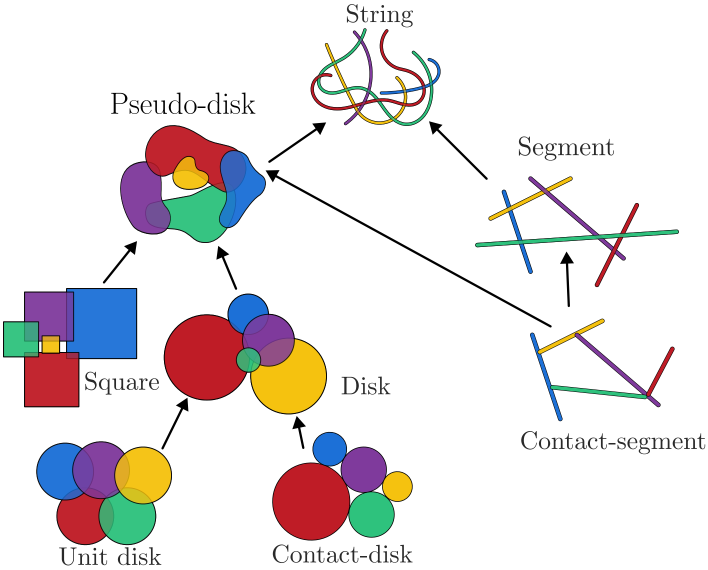


Geometric graphs : Segments in contact





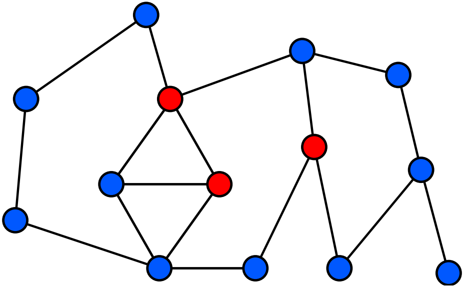
Geometric graphs



The Feedback Vertex Set problem (FVS)

The Feedback vertex set problem

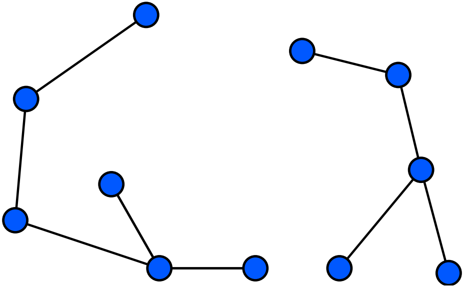
Can we remove at most k vertices of G such that the resulting graph has no cycle ?



The Feedback Vertex Set problem (FVS)

The Feedback vertex set problem

Can we remove at most k vertices of G such that the resulting graph has no cycle ?



The Feedback Vertex Set problem (FVS)

Theorem (Li and Nederlof 2022)

FVS can be solved in time $2.7^k n^{\mathcal{O}(1)}$.

The Feedback Vertex Set problem (FVS)

Theorem (Li and Nederlof 2022)

FVS can be solved in time $2.7^k n^{\mathcal{O}(1)}$.

Theorem

FVS **cannot be solved** in time $2^{o(n)}$ for general graphs (assuming ETH).
Hence no FPT subexponential time algorithm too.

The Feedback Vertex Set problem (FVS)

Theorem (Li and Nederlof 2022)

FVS can be solved in time $2.7^k n^{\mathcal{O}(1)}$.

Theorem

FVS **cannot be solved** in time $2^{o(n)}$ for general graphs (assuming ETH).
Hence no FPT subexponential time algorithm too.

Theorem (Demaine et al. 2005) The Square Root Phenomenon

FVS **in planar graphs** can be solved in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$.
Moreover this is ETH-tight.

The Feedback Vertex Set problem (FVS)

Theorem (Li and Nederlof 2022)

FVS can be solved in time $2.7^k n^{\mathcal{O}(1)}$.

Theorem

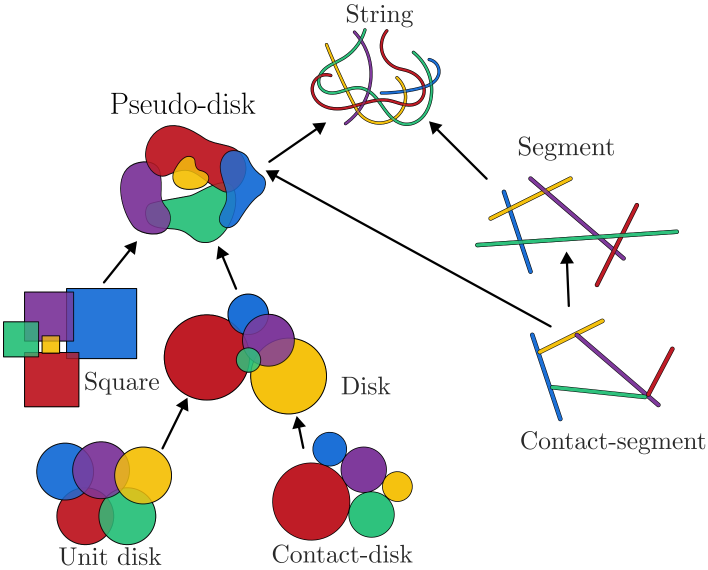
FVS **cannot be solved** in time $2^{o(n)}$ for general graphs (assuming ETH).
Hence no FPT subexponential time algorithm too.

Theorem (Demaine et al. 2005) The Square Root Phenomenon

FVS **in planar graphs** can be solved in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$.
Moreover this is ETH-tight.

What about geometric graph classes ?

The Feedback Vertex Set problem (FVS)



The Feedback Vertex Set problem (FVS)

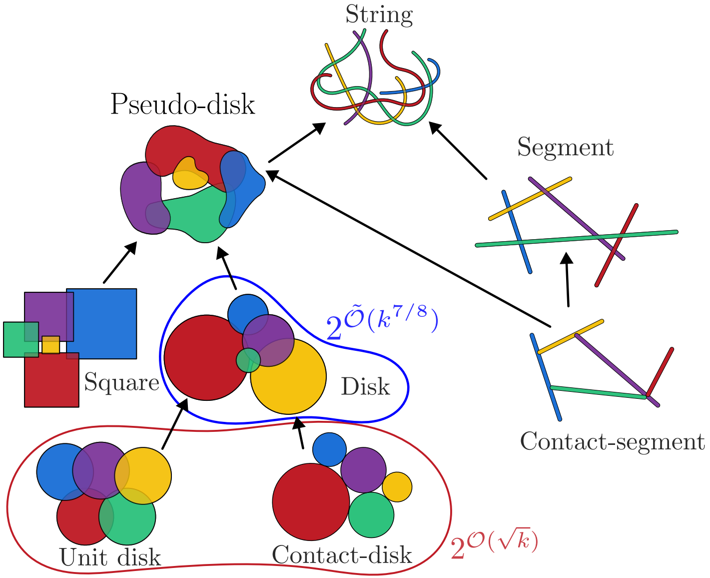
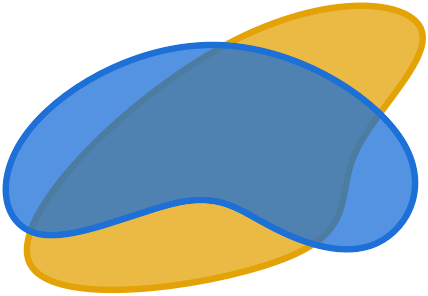


Table of Contents

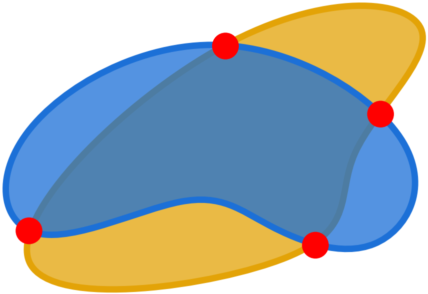
Definition

Not a pseudo-disk :

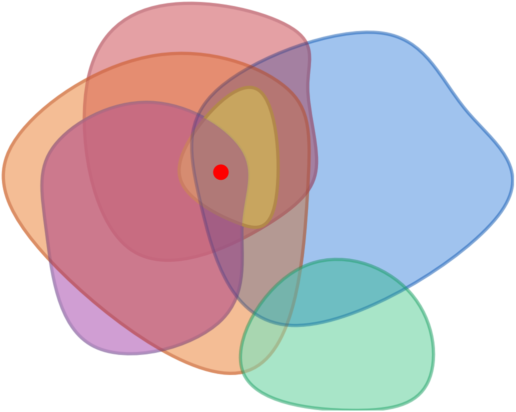


Definition

Not a pseudo-disk :

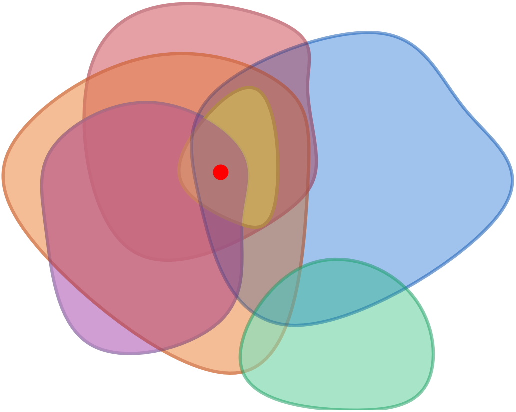


Ply of a set of pseudo-disks



The point is contained in 5 pseudo-disks.

Ply of a set of pseudo-disks

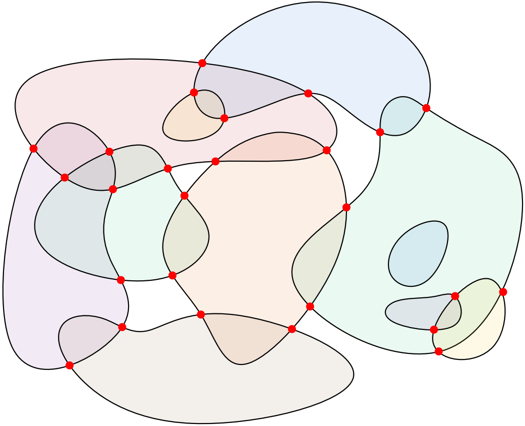


The point is contained in 5 pseudo-disks.
It is maximal : this system of pseudo-disks has ply 5.

Complexity

Theorem (Kedem et al. 1986)

For a set of n pseudo-disks with ply p, the number of regions, arcs and intersection points is $\mathcal{O}(pn)$.



Separators for geometric graphs

Theorem (Lee 2017)

A string graph with m edges has a separator of size $\mathcal{O}(\sqrt{m})$.

Separators for geometric graphs

Theorem (Lee 2017)

A string graph with m edges has a separator of size $\mathcal{O}(\sqrt{m})$.

Lemma (this work)

A pseudo-disk graph with ply p has $\mathcal{O}(pn)$ edges.

Separators for geometric graphs

Theorem (Lee 2017)

A string graph with m edges has a separator of size $\mathcal{O}(\sqrt{m})$.

Lemma (this work)

A pseudo-disk graph with ply p has $\mathcal{O}(pn)$ edges.

Corollary

For a pseudo-disk graph G with ply p, we have $\mathrm{tw}(G) = \mathcal{O}(\sqrt{pn})$.

Separators for geometric graphs

Theorem (Lee 2017)

A string graph with m edges has a separator of size $\mathcal{O}(\sqrt{m})$.

Lemma (this work)

A pseudo-disk graph with ply p has $\mathcal{O}(pn)$ edges.

Corollary

For a pseudo-disk graph G with ply p, we have $\mathrm{tw}(G) = \mathcal{O}(\sqrt{pn})$.

Theorem (Cygan et al. 2011)

FVS can be solved in time $2^{\mathcal{O}(\mathrm{tw}(G))} n^{\mathcal{O}(1)}$.

Table of Contents

The main steps of the algorithm are :

1. Compute a small set M of "important" vertices.

Overview of the algorithm

The main steps of the algorithm are :

1. Compute a small set M of "important" vertices.
2. Reduce the number of vertices outside M.

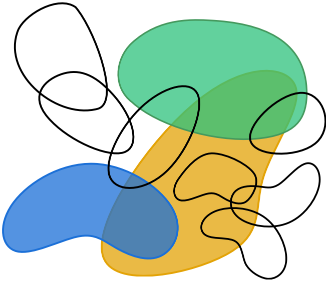
Overview of the algorithm

The main steps of the algorithm are :

1. Compute a small set M of "important" vertices.
2. Reduce the number of vertices outside M.
3. Bound the treewidth of the obtained kernel.

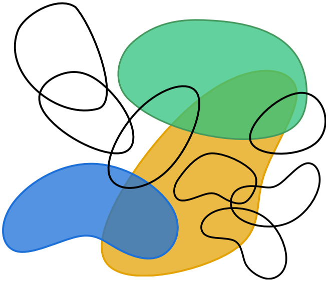
The set M

- M_0: 2-approximation of the solution. $|M_0| = \mathcal{O}(k)$.



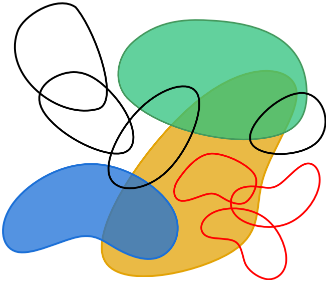
The set M

- M_0: 2-approximation of the solution. $|M_0| = \mathcal{O}(k)$.
- M_1: add vertices such that for all $v \in M_1$, $N_G(v) \setminus M_1$ is an independant set (using a branching method from Lokshtanov et al.). $|M_1| = \mathcal{O}(pk)$.



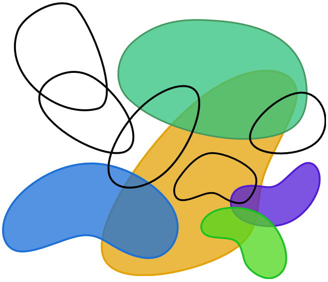
The set M

- M_0: 2-approximation of the solution. $|M_0| = \mathcal{O}(k)$.
- M_1: add vertices such that for all $v \in M_1$, $N_G(v) \setminus M_1$ is an independant set (using a branching method from Lokshtanov et al.). $|M_1| = \mathcal{O}(pk)$.



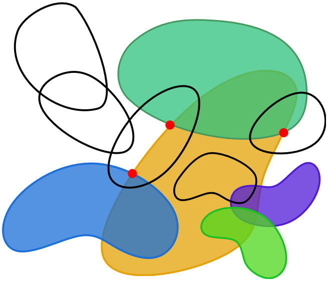
The set M

- M_0: 2-approximation of the solution. $|M_0| = \mathcal{O}(k)$.
- M_1: add vertices such that for all $v \in M_1$, $N_G(v) \setminus M_1$ is an independant set (using a branching method from Lokshtanov et al.). $|M_1| = \mathcal{O}(pk)$.



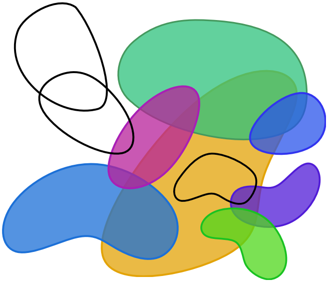
The set M

- M_0: 2-approximation of the solution. $|M_0| = \mathcal{O}(k)$.
- M_1: add vertices such that for all $v \in M_1$, $N_G(v) \setminus M_1$ is an independant set (using a branching method from Lokshtanov et al.). $|M_1| = \mathcal{O}(pk)$.
- M: add the pseudo-disks containing an intersection point of two pseudo-disks of M_1. $|M| = \mathcal{O}(p^2 k)$.

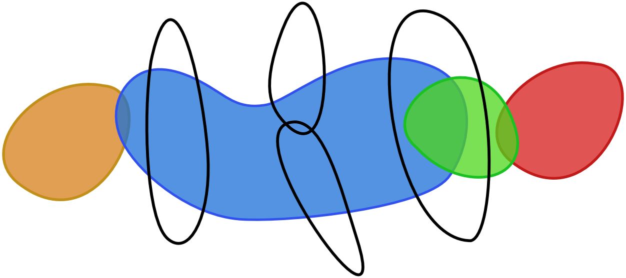


The set M

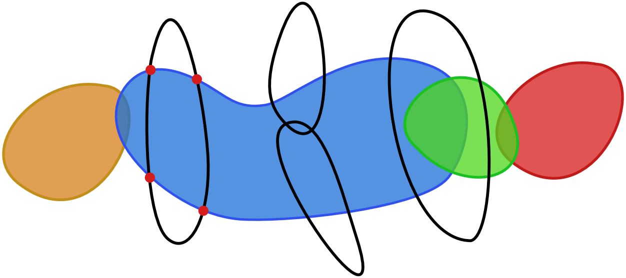
- M_0: 2-approximation of the solution. $|M_0| = \mathcal{O}(k)$.
- M_1: add vertices such that for all $v \in M_1$, $N_G(v) \setminus M_1$ is an independant set (using a branching method from Lokshtanov et al.). $|M_1| = \mathcal{O}(pk)$.
- M: add the pseudo-disks containing an intersection point of two pseudo-disks of M_1. $|M| = \mathcal{O}(p^2k)$.



The set M cannot be "crossed"

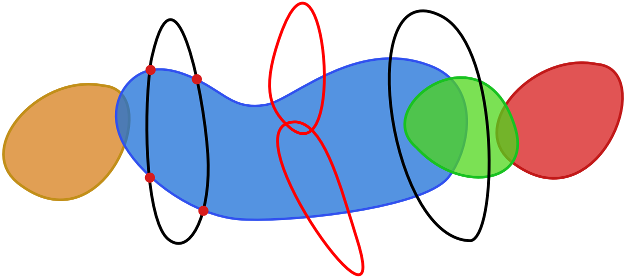


The set M cannot be "crossed"



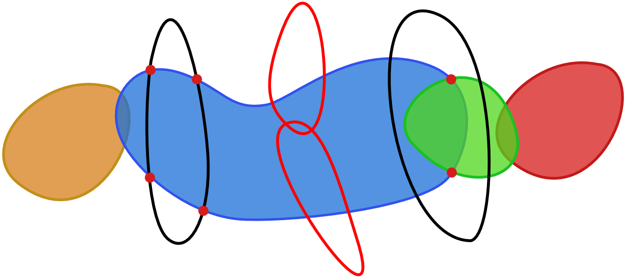
Not a system of pseudo-disks !

The set M cannot be "crossed"



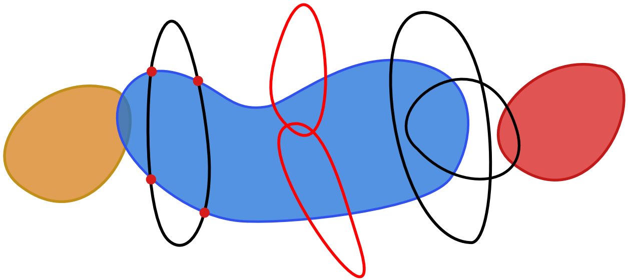
The neighbors of Blue outside M are not independant.

The set M cannot be "crossed"



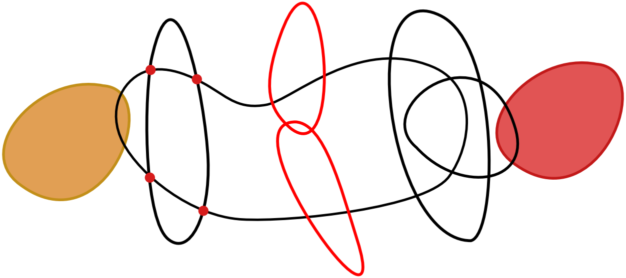
Contains intersection points of pseudo-disks in M.
Either Green or Blue was not in M_1.

The set M cannot be "crossed"



If Blue $\in M_1$ and Green $\notin M_1$,
the neighbors of Blue outside M_1 are not independant.

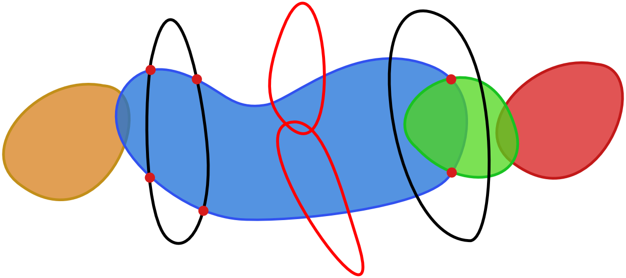
The set M cannot be "crossed"



A triangle outside of M_1.

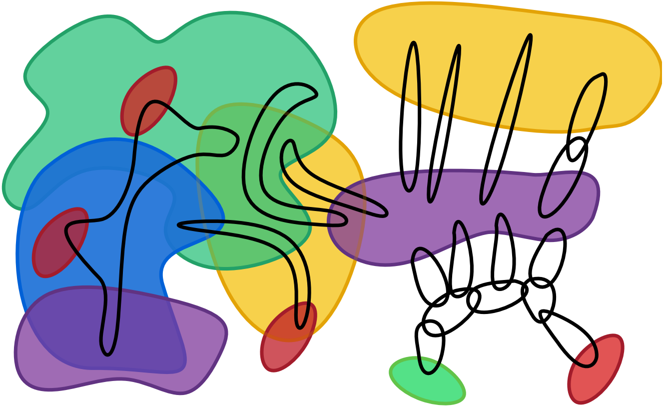
Impossible !

The set M cannot be "crossed"

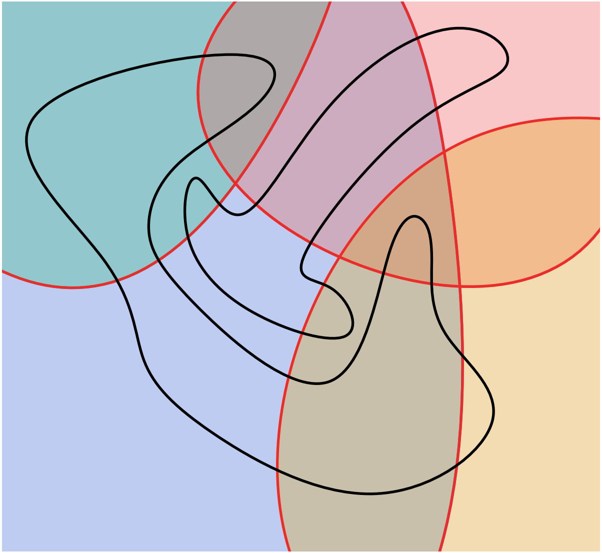


We have no crossing of M,
and the pseudo-disks outside M do not contain intersection points of M.

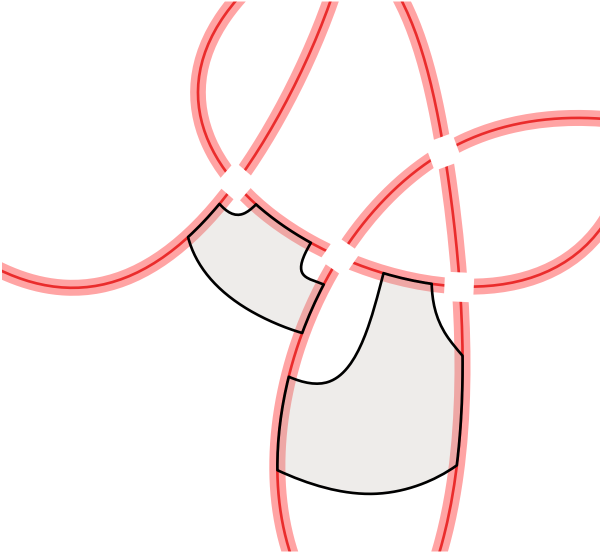
What remains outside of M



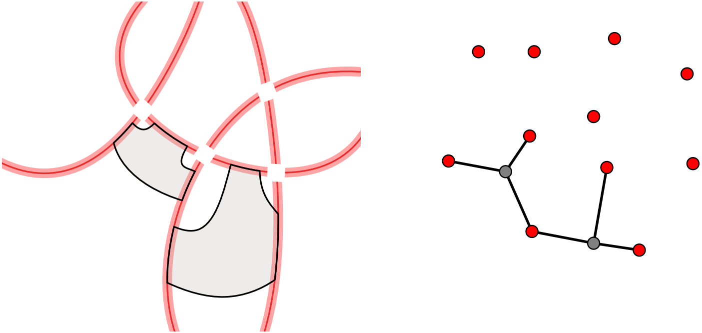
Inner vertices : the case of pseudo-disks with "branching"



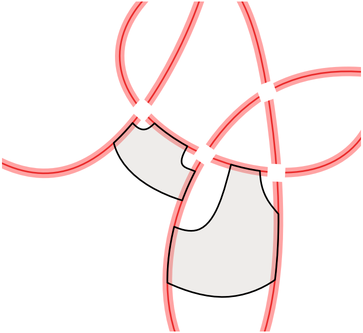
Inner vertices : the case of pseudo-disks with "branching"

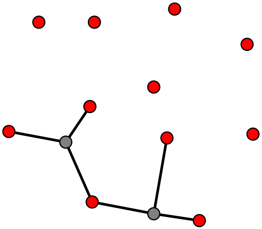


Inner vertices : the case of pseudo-disks with "branching"



Inner vertices : the case of pseudo-disks with "branching"

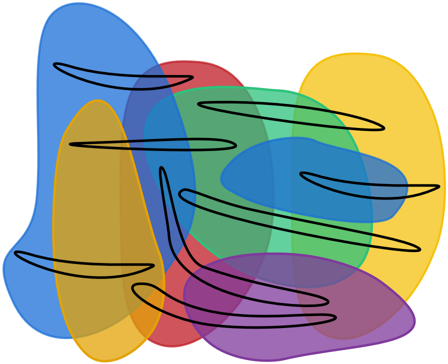




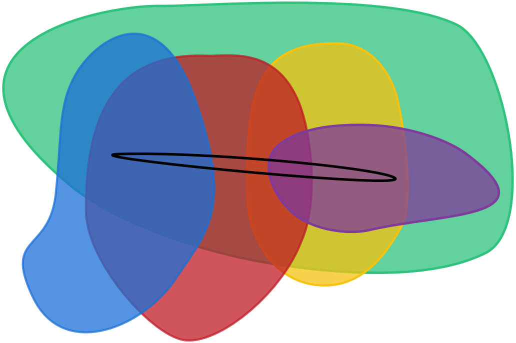
Lemma

For a planar bipartite graph with parts A and B such that for all $v \in A$, $\Delta(v) \geq 3$, we have $|A| = \mathcal{O}(|B|)$.

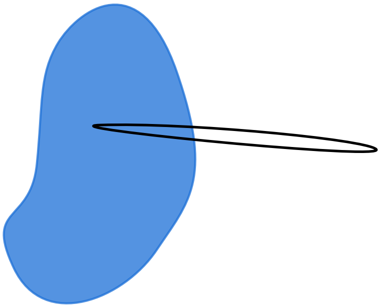
Inner vertices: the case of paths



Inner vertices: the case of paths

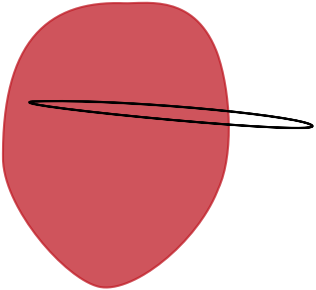


Inner vertices: the case of paths



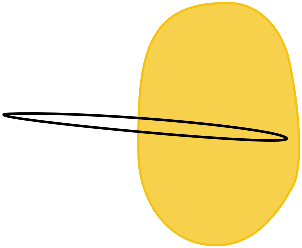
All the neighbors contain at least one endpoint of the path.

Inner vertices: the case of paths



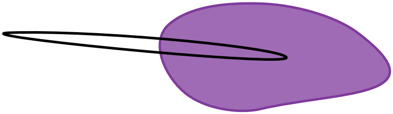
All the neighbors contain at least one endpoint of the path.

Inner vertices: the case of paths



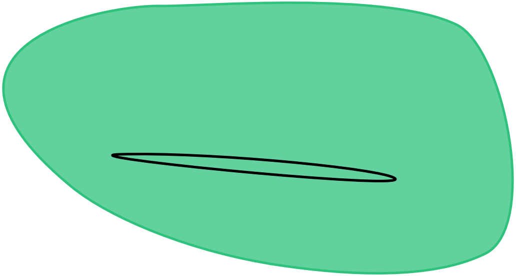
All the neighbors contain at least one endpoint of the path.

Inner vertices: the case of paths



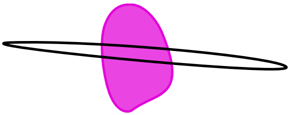
All the neighbors contain at least one endpoint of the path.

Inner vertices: the case of paths



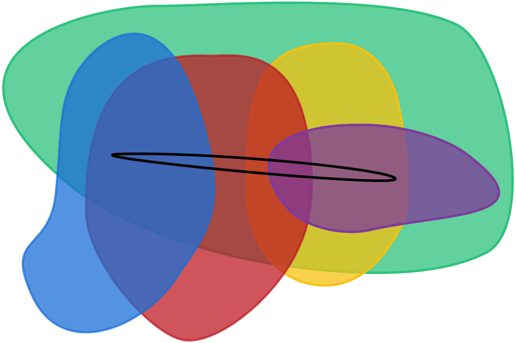
All the neighbors contain at least one endpoint of the path.

Inner vertices: the case of paths



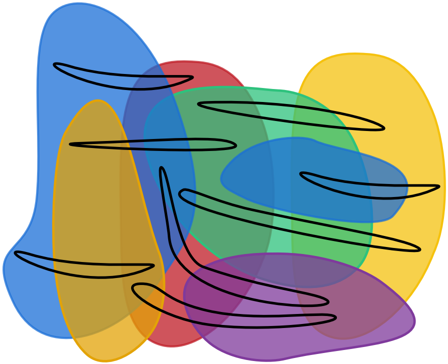
Impossible !

Inner vertices: the case of paths



The neighborhood of a path is the union of the 2 cliques of it extremities.
So it has at most $2p$ neighbors.

Inner vertices: the case of paths



Theorem (Aronov et al. 2020, Keszegh 2018)

Given two sets of pseudo-disks $\mathcal{M}$ and $\mathcal{P}$ where each element of $\mathcal{P}$ intersects at most k elements in $\mathcal{M}$, then the elements of $\mathcal{P}$ have at most $\mathcal{O}(k^3|\mathcal{M}|)$ distinct neighborhoods in $\mathcal{M}$.

A first kernelization rule : removing twins

Kernelization rule

Given a set X and $|X| + 2$ independant false twins with neighborhood X, removing one of the twins does not change the size of the optimal FVS.

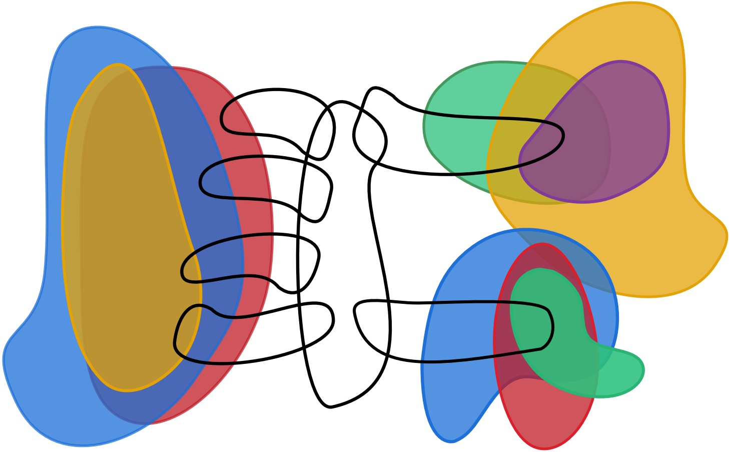
A first kernelization rule : removing twins

Kernelization rule

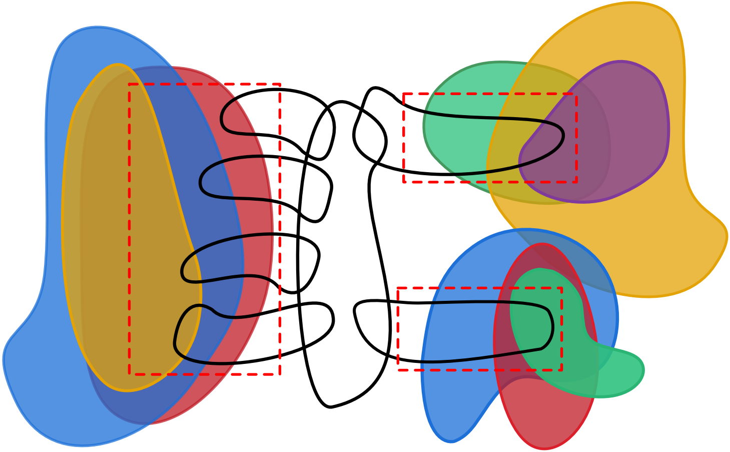
Given a set X and $|X| + 2$ independant false twins with neighborhood X, removing one of the twins does not change the size of the optimal FVS.

Stronger result : In the case where X is the union of two cliques, having 6 twins is enough to ensure that removing a twin is safe.

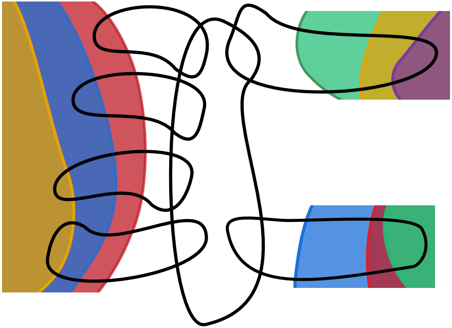
Remaining trees : chunks



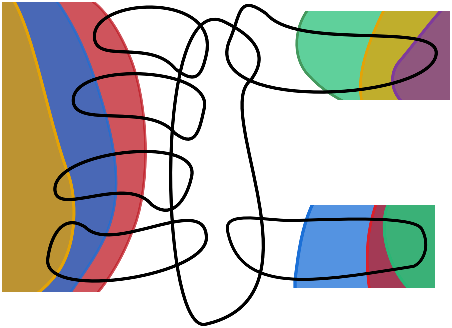
Remaining trees : chunks



Remaining trees : chunks



Remaining trees : chunks



$$\#Chunks(T) = 3.$$

Remaining trees

T a tree of $G - M$.
$\#Chunks(T)$: Number of distinct chunks visited by the tree T.

Remaining trees

T a tree of $G - M$.
$\#Chunks(T)$: Number of distinct chunks visited by the tree T.

- The number and the size of the trees with $\#Chunks(T) \leq 1$ are bounded by kernelization rules.

Remaining trees

T a tree of $G - M$.
$\#Chunks(T)$: Number of distinct chunks visited by the tree T.

- The number and the size of the trees with $\#Chunks(T) \leq 1$ are bounded by kernelization rules.
- For trees with $\#Chunks(T) = 2$, they can be seen as edges of a planar graph with $p|M|$ vertices.

Remaining trees

T a tree of $G - M$.
$\#Chunks(T)$: Number of distinct chunks visited by the tree T.

- The number and the size of the trees with $\#Chunks(T) \leq 1$ are bounded by kernelization rules.
- For trees with $\#Chunks(T) = 2$, they can be seen as edges of a planar graph with $p|M|$ vertices.
- For trees with $\#Chunks(T) \geq 3$, we can again define a bipartite planar graph.

The case $\#Chunks(T) \geq 3$

Lemma

$$\sum_{\substack{T \in G-M \\ \#Chunks(T) \geq 3}} \#Chunks(T) = \mathcal{O}(p|M|).$$

The case $\#Chunks(T) \geq 3$

Lemma

$$\sum_{\substack{T \in G-M \\ \#Chunks(T) \geq 3}} \#Chunks(T) = \mathcal{O}(p|M|).$$

Goal

Bound the number of vertices of a tree $T \in G - M$ relatively to $\#Chunks(T)$.

The case $\#Chunks(T) \geq 3$

Lemma

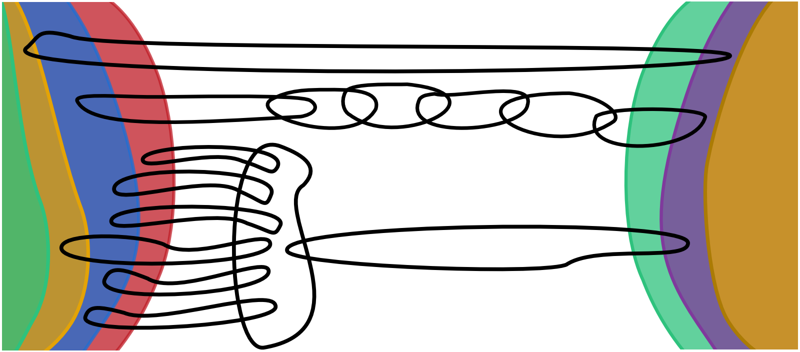
$$\sum_{\substack{T \in G-M \\ \#Chunks(T) \geq 3}} \#Chunks(T) = \mathcal{O}(p|M|).$$

Goal

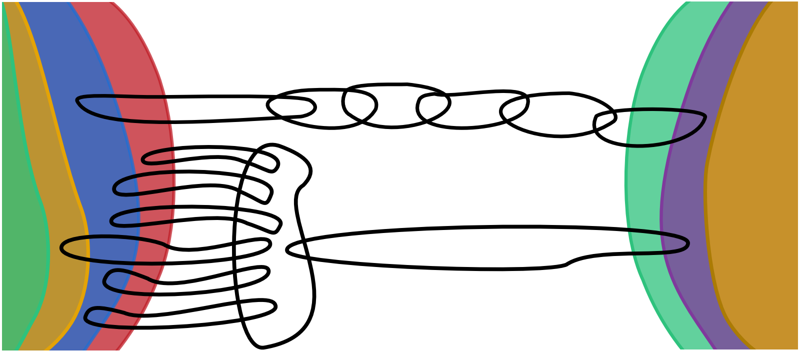
Bound the number of vertices of a tree $T \in G - M$ relatively to $\#Chunks(T)$.

Done by multiple kernelization rules.

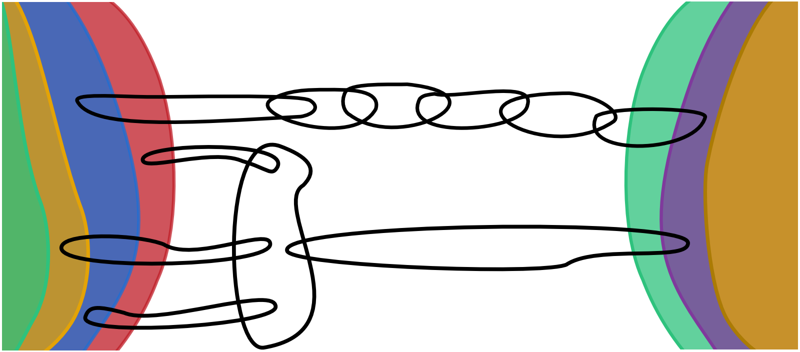
Example of kernelization



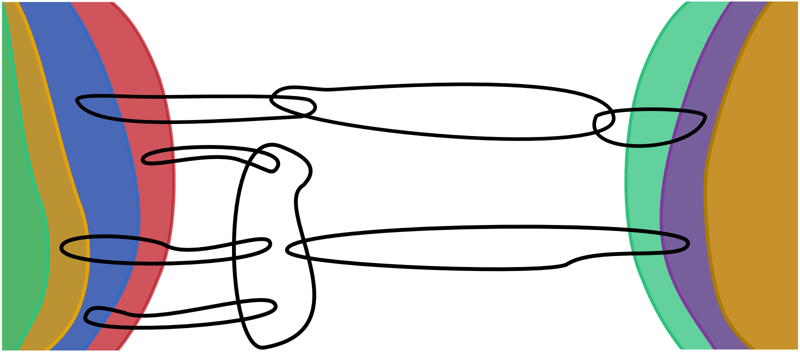
Example of kernelization



Example of kernelization



Example of kernelization



Bounding the treewidth

Lemma

FVS for pseudo-disks of ply p can be reduced to $2^{\tilde{\mathcal{O}}(k^{1-\epsilon})}$ FVS instances (G', k') with $k' \leq k$ and G' pseudo-disk graphs with $\mathcal{O}(kp^4)$ vertices.

Bounding the treewidth

Lemma

FVS for pseudo-disks of ply p can be reduced to $2^{\tilde{\mathcal{O}}(k^{1-\epsilon})}$ FVS instances (G', k') with $k' \leq k$ and G' pseudo-disk graphs with $\mathcal{O}(kp^4)$ vertices.

By guessing which vertices of the big cliques are in the solution, we get $p = \tilde{\mathcal{O}}(k^{\epsilon})$ for a price $2^{\tilde{\mathcal{O}}(k^{1-\epsilon})}$.

Bounding the treewidth

Lemma

FVS for pseudo-disks of ply p can be reduced to $2^{\tilde{\mathcal{O}}(k^{1-\epsilon})}$ FVS instances (G', k') with $k' \leq k$ and G' pseudo-disk graphs with $\mathcal{O}(kp^4)$ vertices.

By guessing which vertices of the big cliques are in the solution, we get $p = \tilde{\mathcal{O}}(k^{\epsilon})$ for a price $2^{\tilde{\mathcal{O}}(k^{1-\epsilon})}$.

The generated instances have $\mathrm{tw}(G') = \mathcal{O}(\sqrt{pn}) = \mathcal{O}(k^{1/2+2.5\epsilon})$.

Bounding the treewidth

Lemma

FVS for pseudo-disks of ply p can be reduced to $2^{\tilde{\mathcal{O}}(k^{1-\epsilon})}$ FVS instances (G', k') with $k' \leq k$ and G' pseudo-disk graphs with $\mathcal{O}(kp^4)$ vertices.

By guessing which vertices of the big cliques are in the solution, we get $p = \tilde{\mathcal{O}}(k^{\epsilon})$ for a price $2^{\tilde{\mathcal{O}}(k^{1-\epsilon})}$.

The generated instances have $\text{tw}(G') = \mathcal{O}(\sqrt{pn}) = \mathcal{O}(k^{1/2+2.5\epsilon})$.

Theorem (this work)

Given a representation, FVS can be solved in time $2^{\tilde{\mathcal{O}}(k^{6/7})} n^{\mathcal{O}(1)}$ in pseudo-disk graphs.

The new cases

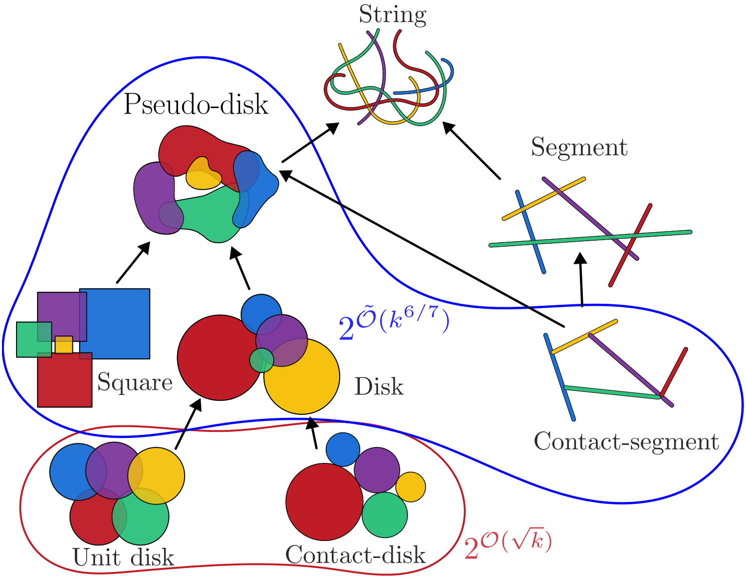


Table of Contents

Conclusion

Our result :

- $2^{o(k)} n^{\mathcal{O}(1)}$ algorithm for FVS for pseudo-disks.

Conclusion

Our result :

- $2^{o(k)} n^{\mathcal{O}(1)}$ algorithm for FVS for pseudo-disks.

Possible next steps :

- Going toward intersection of segments.

Conclusion

Our result :

- $2^{o(k)} n^{\mathcal{O}(1)}$ algorithm for FVS for pseudo-disks.

Possible next steps :

- Going toward intersection of segments.
- Intersection of fat objects.

Conclusion

Our result :

- $2^{o(k)} n^{\mathcal{O}(1)}$ algorithm for FVS for pseudo-disks.

Possible next steps :

- Going toward intersection of segments.
- Intersection of fat objects.
- Exponential lower bound for string graphs (assuming ETH) ?

Conclusion

Our result :

- $2^{o(k)} n^{\mathcal{O}(1)}$ algorithm for FVS for pseudo-disks.

Possible next steps :

- Going toward intersection of segments.
- Intersection of fat objects.
- Exponential lower bound for string graphs (assuming ETH) ?

Thanks for your attention !