Linear kernel for planar Dominating Set

Input: (G, k), G is planar

Ouput: (G', k') s.t.

- (G, k) is a yes-instance if and only if (G', k') is a yes-instance,

Linear kernel for planar Dominating Set

Input: (G, k), G is planar

Ouput: (G', k') s.t.

- (G, k) is a yes-instance if and only if (G', k') is a yes-instance,
- G' is planar (actually a subgraph of G), and

Linear kernel for planar Dominating Set

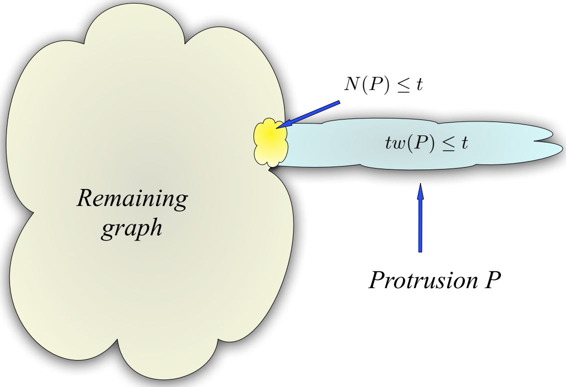
Input: (G, k), G is planar

Ouput: (G', k') s.t.

- (G, k) is a yes-instance if and only if (G', k') is a yes-instance,
- G' is planar (actually a subgraph of G), and
- $|G| = \mathcal{O}(k)$.

Tool

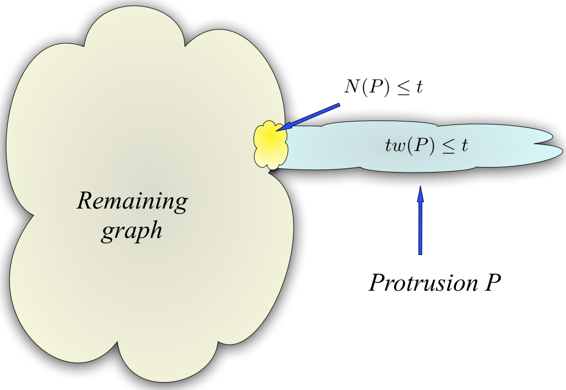
Protrusion in planar graph



Tool

Reducing Protrusion

Reduce the number of vertices in P down to $\mathcal{O}(|OPT \cap P|)$.



Protrusion Reduction Rule

- For every partial solution D of $N(P)$ compute a minimum extension of D to $P \cup N(P)$. (Partial solution: in dominating set, is dominated and not dominated.) We have at most 3^t partial solutions for each P .

Protrusion Reduction Rule

- For every partial solution D of $N(P)$ compute a minimum extension of D to $P \cup N(P)$. (Partial solution: in dominating set, is dominated and not dominated.) We have at most 3^t partial solutions for each P .
- If the size of the solution is at most $|OPT(G) \cap P| + t$, mark the vertices of the solution

Protrusion Reduction Rule

- For every partial solution D of $N(P)$ compute a minimum extension of D to $P \cup N(P)$. (Partial solution: in dominating set, is dominated and not dominated.) We have at most 3^t partial solutions for each P .
- If the size of the solution is at most $|OPT(G) \cap P| + t$, mark the vertices of the solution
- Delete edges with both unmarked endpoints. (If there is an optimal solution in G, then there is an optimal solution using only marked vertices.)

Protrusion Reduction Rule

- For every partial solution D of $N(P)$ compute a minimum extension of D to $P \cup N(P)$. (Partial solution: in dominating set, is dominated and not dominated.) We have at most 3^t partial solutions for each P .
- If the size of the solution is at most $|OPT(G) \cap P| + t$, mark the vertices of the solution
- Delete edges with both unmarked endpoints. (If there is an optimal solution in G, then there is an optimal solution using only marked vertices.)
- Delete isolated vertices.

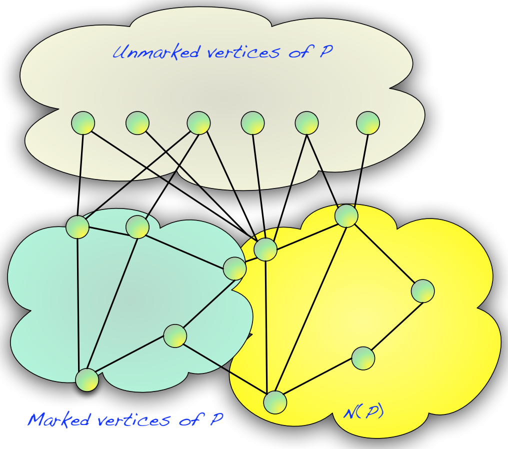
Protrusion Reduction Rule

How many marked vertices in P?

- $3^t(|OPT(G) \cap P| + t)$

Reduction Rule for unmarked vertices

How many unmarked vertices in P?



Planar Bipartite Lemma

Lemma (Planar Bipartite Lemma)

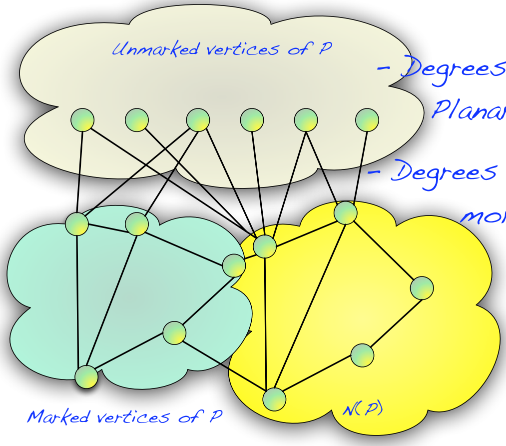
Let G be a simple bipartite graph with a bipartition (A, B) such that every vertex $v \in B$ has at least three neighbors in A. Then $|B| \leq 2|A|$

Proof: Euler's formula $m = n + f - 2$.

Reduction Rule for unmarked vertices

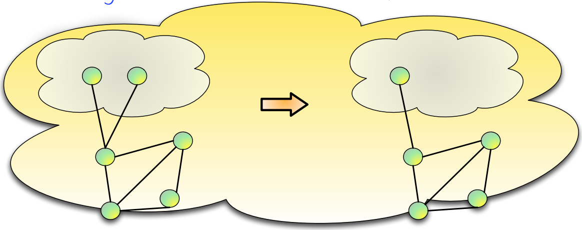
How many unmarked vertices in P?

- Degrees >= 3: constant (again Planar Bipartite Lemma)
- Degrees 1 and 2 can be many, need more reduction rules

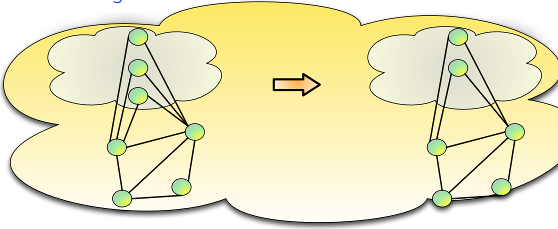


Reduction Rule for unmarked vertices

Degree-1 Reduction Rule for unmarked vertices



Degree-2 Reduction Rule for unmarked vertices



Reduction Rule for unmarked vertices

After application of Reduction Rules, what remains from P has $3^t(|OPT(G) \cap P| + t)$ marked and $\mathcal{O}(3^t(|OPT(G) \cap P| + t))$ unmarked vertices.

Strategy

- Decompose G into protrusions

Strategy

- Decompose G into protrusions
- Use protrusion reduction rule

Decompositing G

Input: (G, k), G is planar (and connected)

Compute a connected dominating set S of size $\mathcal{O}(k)$.

Claim

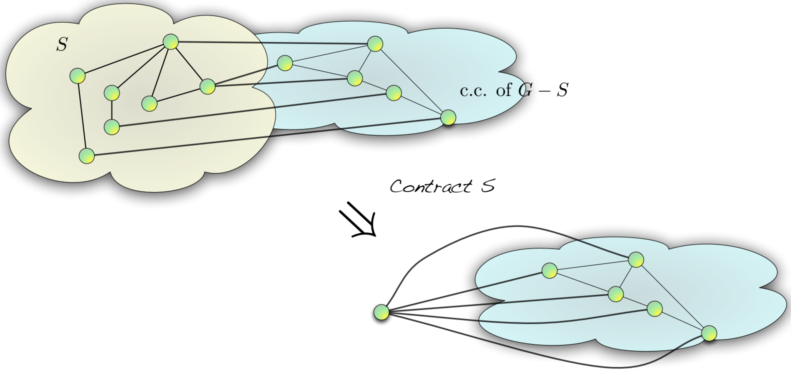
Every connected component of $G - S$ is an outerplanar graph.

Decomposing G

Claim

Every connected component of $G - S$ is an outerplanar graph.

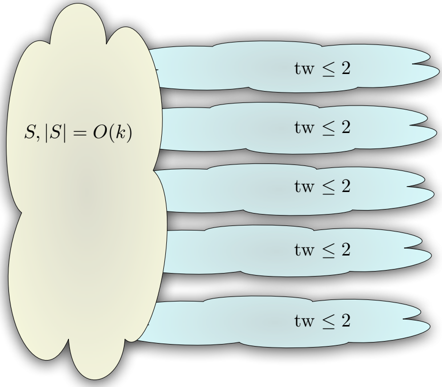
Proof:



Decomposing G

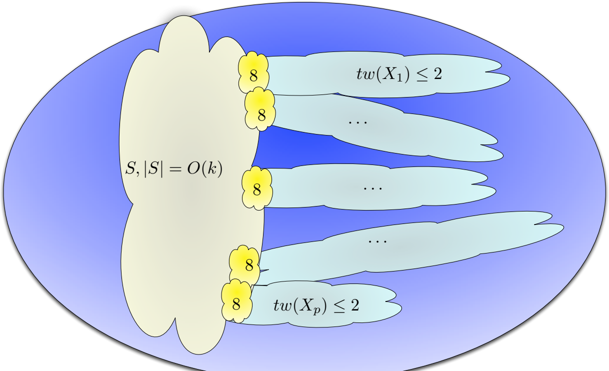
We have set S of size $\mathcal{O}(k)$ and every connected component of $G - S$ is of treewidth at most 2.

But the neighborhoods of components can be large.

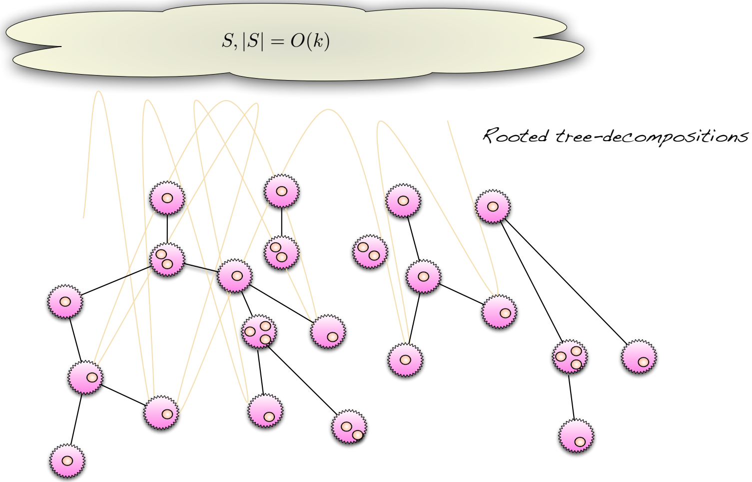


Decomposing G

We have set S of size $\mathcal{O}(k)$ and every connected component of $G - S$ is of treewidth at most 2. We want (a little bit) more. We want to enhance S such that S is still of order $\mathcal{O}(k)$ but for every c.c. X of $G - S$, $|N_G(X)| \leq 8$.



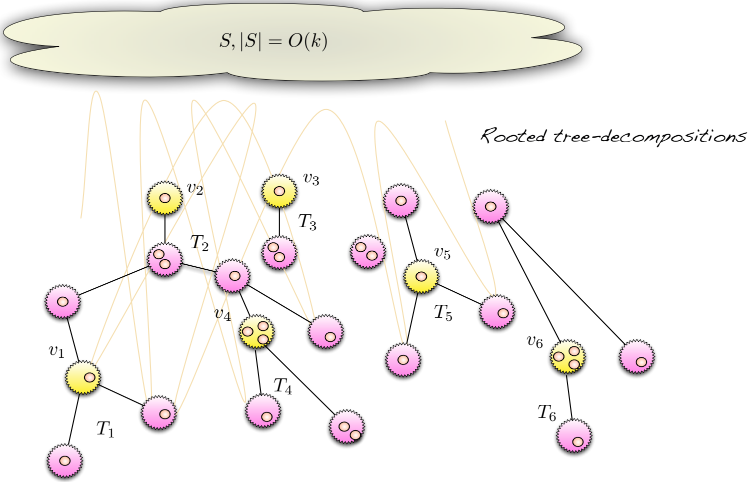
Cutting components



Marking step

- Set marked vertices $M = \emptyset$, $i = 1$
- While a component C of $G - (S \cup M)$ has ≥ 3 neighbors in S **do**
  - Let v_i be the leftmost bag in the tree decomposition T of C such that the vertices contained in bags of the subtree T_i rooted in v_i have ≥ 3 neighbors in S
  - Add to M vertices contained in v_i
  - $i := i + 1$, delete vertices contained in nodes of T_i from G.

Cutting components



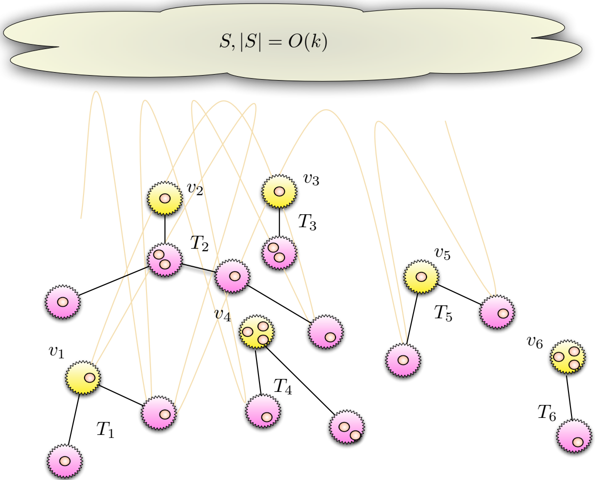
How many vertices are marked? Again

Lemma (Planar Bipartite Lemma)

Let G be a simple bipartite graph with a bipartition (A, B) such that every vertex $v \in B$ has at least three neighbors in A. Then $|B| \leq 2|A|$

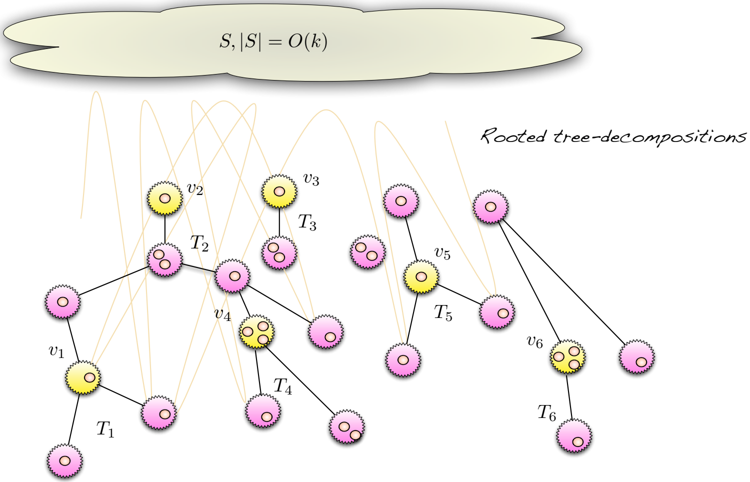
How many vertices are marked?

By Planar Bipartite Lemma, $i \leq 2|S|$. Each node v_i contains at most 3 vertices, hence $|M| \leq 6|S|$.



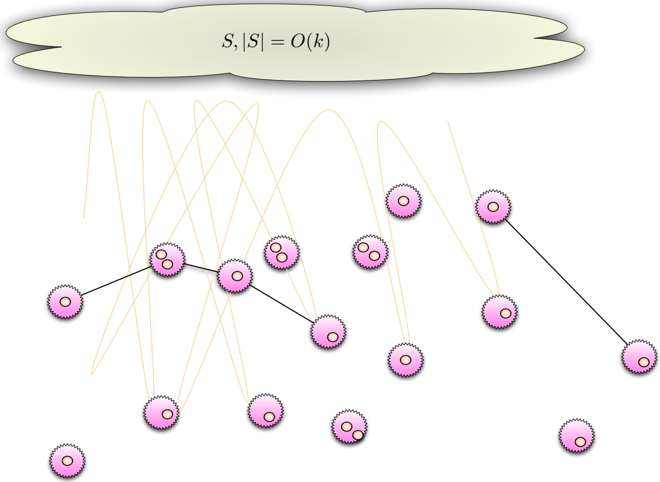
Cutting components

Every c.c. of $G - (S \cup M)$ sees at most 2 vertices in S.



Cutting components

Every c.c. of $G - (S \cup M)$ sees at most 2 vertices in S.



We are almost there!

Every c.c. of $G - (S \cup M)$ sees at most 2 vertices in S but can see many vertices in M.

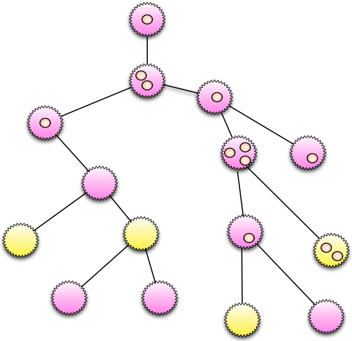
We are almost there!

Every c.c. of $G - (S \cup M)$ sees at most 2 vertices in S but can see many vertices in M.

Mark more vertices!

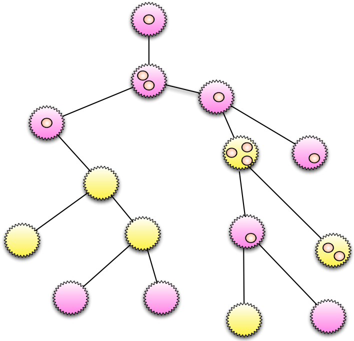
Least common ancestor closure

While there are two marked nodes whose common ancestor is not marked, mark it.



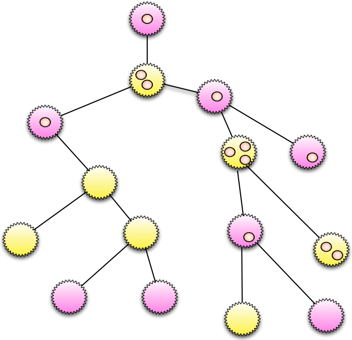
Least common ancestor closure

While there are two marked nodes whose common ancestor is not marked, mark it.



Least common ancestor closure

While there are two marked nodes whose common ancestor is not marked, mark it.



Least common ancestor closure

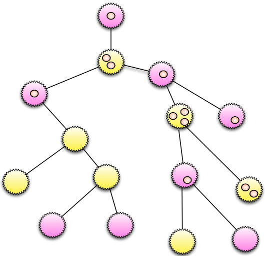
- By standard tree arguments, we marked at most $|M|$ additional vertices

Least common ancestor closure

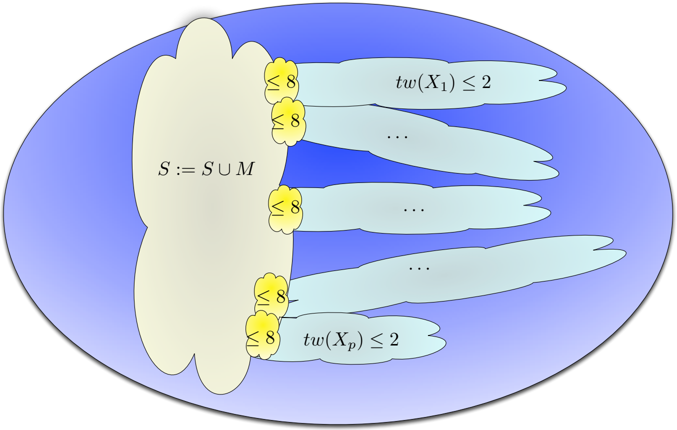
- By standard tree arguments, we marked at most $|M|$ additional vertices
- Put all marked vertices in M. The size of $S \cup M$ is now at most $13|S|$

Least common ancestor closure

- By standard tree arguments, we marked at most $|M|$ additional vertices
- Put all marked vertices in M. The size of $S \cup M$ is now at most $13|S|$
- Every connected component of $G - (S \cup M)$ sees at most 2 vertices in S and at most $2 \cdot 3 = 6$ vertices in M



Final decomposition



Different neighborhoods

Group components of $G - S$ into classes according to their neighborhoods in S.

Different neighborhoods

Group components of $G - S$ into classes according to their neighborhoods in S. How many classes?

Different neighborhoods

Group components of $G - S$ into classes according to their neighborhoods in S. How many classes?

- Number of different size-1 neighborhoods is at most $|S|$
- Number of different size-2 neighborhoods is at most $3|S|$ (Euler formula)
- Number of different size-≥ 3 neighborhoods is at most $2|S|$ (Planar Bipartite Lemma)

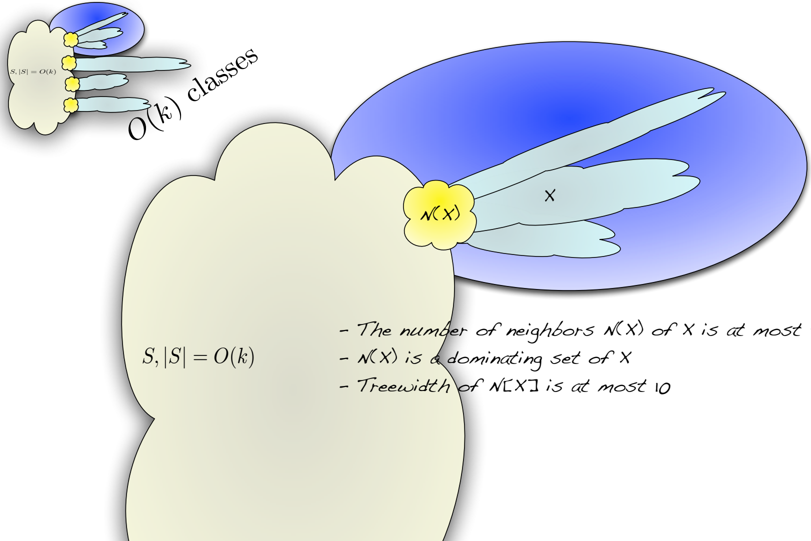
Different neighborhoods

Group components of $G - S$ into classes according to their neighborhoods in S. How many classes?

- Number of different size-1 neighborhoods is at most $|S|$
- Number of different size-2 neighborhoods is at most $3|S|$ (Euler formula)
- Number of different size-≥ 3 neighborhoods is at most $2|S|$ (Planar Bipartite Lemma)

In total, at most $6|S|$ classes.

Closer look at a class



- The number of neighbors N(X) of X is at most 8
- N(X) is a dominating set of X
- Treewidth of N[X] is at most 10

Reduction Rule

- Each class X is a protrusion, so we can apply Protrusion Reduction Rule

Reduction Rule

- Each class X is a protrusion, so we can apply Protrusion Reduction Rule
- Since for each X, $|OPT(G) \cap X| \leq 8$, each protrusion shrinks to constant size.

Recap of the algorithm

- Find a set S of size $O(k)$ s.t. $G - S$ is of constant treewidth

- Find a set M of size $O(k)$ s.t. every c.c. of $G-(S \cup M)$ sees constant number of vertices in $S \cup M$

- Group c.c. of $G-(S \cup M)$ into classes according to their neighborhoods

- Reduce each class to constant size

Where the property of a dominating set was important?

✓ - Find a set S of size $O(k)$ s.t. $G - S$ is of constant treewidth

- Find a set M of size $O(k)$ s.t. every c.c. of $G-(S \cup M)$ sees constant number of vertices in $S \cup M$

- Group c.c. of $G-(S \cup M)$ into classes according to their neighborhoods

✓ - Reduce each class to constant size

Abstraction of the property

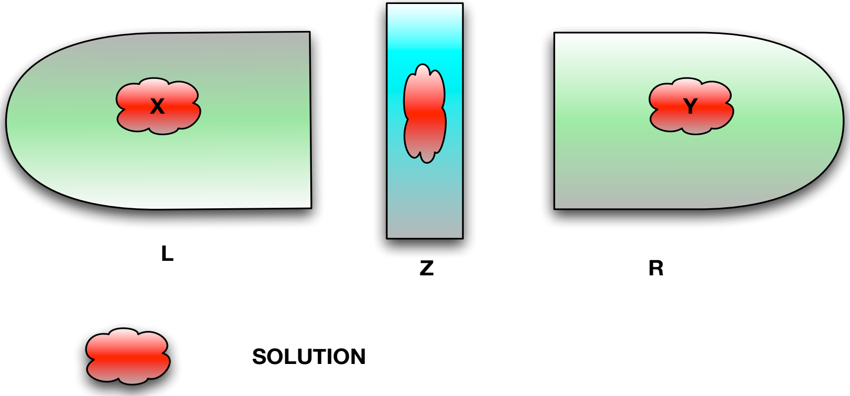
Parameterized problem (G,k) such that:

- If (G,k) is a YES instance, we can find a set S of size $O(k)$ s.t. $G - S$ is of constant treewidth

- We can reduce each class to constant size

WE HAVE LINEAR KERNEL!

Concept of Separability



$$|OPT(L \cup Z)| = |OPT(G) \cap L| \pm \mathcal{O}(|Z|)$$

$$|OPT(R \cup Z)| = |OPT(G) \cap R| \pm \mathcal{O}(|Z|)$$

Bidimensional Problems

- Minor Free Graph Class + Minor Bidimensionality

$$\implies \mathrm{tw}(\mathcal{O}(\sqrt{k}))$$

(Sublinear Treewidth Parameter Bound)

- APEX-Minor Free Graph Class + Contraction Bidimensionality

$$\implies \mathrm{tw}(\mathcal{O}(\sqrt{k}))$$

Bidimensional Problems

- Minor Free Graph Class + Minor Bidimensionality

$$\implies \mathrm{tw}(\mathcal{O}(\sqrt{k}))$$

(Sublinear Treewidth Parameter Bound)

- APEX-Minor Free Graph Class + Contraction Bidimensionality

$$\implies \mathrm{tw}(\mathcal{O}(\sqrt{k}))$$

- Separability + Sublinear Treewidth Parameter Bound

$$\implies \exists\, S \text{ of size } \mathcal{O}(k) \text{ such that } \quad \mathrm{tw}(G[V \setminus S]) \leq \eta$$

(η is a constant that only depends on the problem)

Treewidth modulator

Main Idea: All Bidimensional and Seperable problems are

"constant factor" related to:

Treewidth η-Modulator

Input: A graph $G = (V, E)$ and a positive integers k.

Parameter: k

Question: Does there exist a subset $F \subseteq V$ of size at most k such that $\mathrm{tw}(G[V \setminus F]) \leq \eta$?

on "planar like graphs".

Approximating treewidth modulator

Constant factor approximation:

FVF, Daniel Lokshtanov, Neeldhara Misra, Saket Saurabh

Planar F-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. FOCS 2012: 470-479

Approximating treewidth modulator

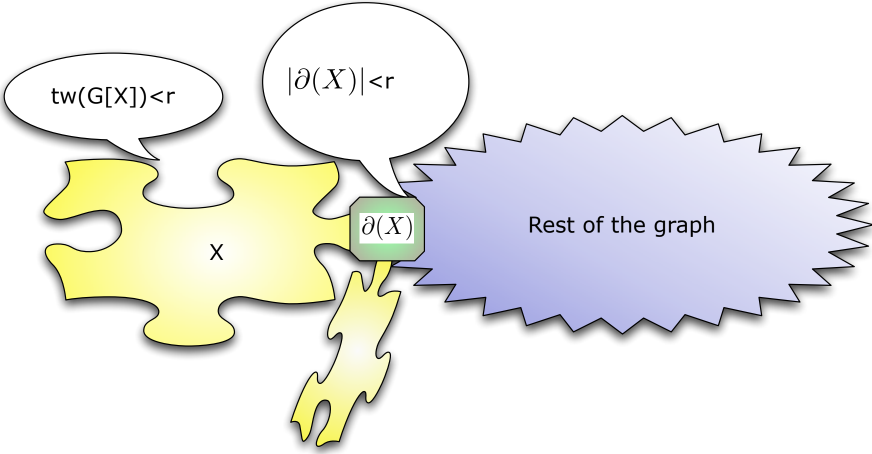
Constant factor approximation:

FVF, Daniel Lokshtanov, Neeldhara Misra, Saket Saurabh

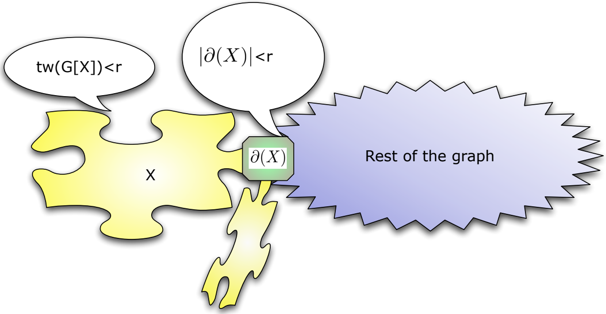
Planar F-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. FOCS 2012: 470-479

Thus for every instance (G, k) of a bidimensional separable problem on planar (apex-minor-free, minor-free) one can find in polynomial time a set S of size $\mathcal{O}(k)$ such that $G - S$ is of constant treewidth.

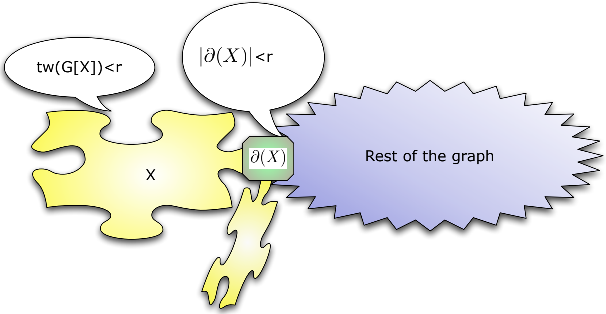
Reduction rules and protrusions



Protrusions



Protrusions

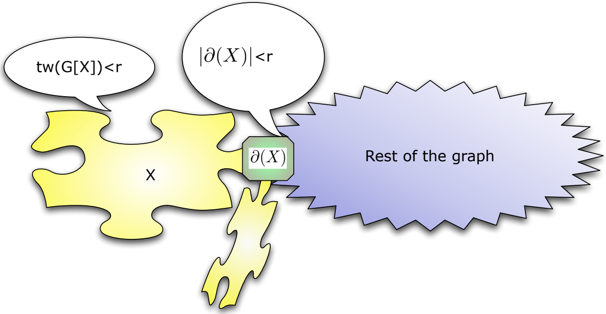


Idea to use this object in kernelization:

Bodlaender, FF, Lokshtanov, Penninkx, Saurabh, Thilikos

(Meta) Kernelization. FOCS 2009: 629-638

Protrusions



Idea to use this object in kernelization:

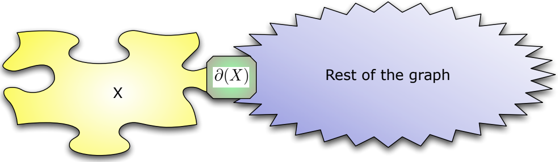
Bodlaender, FF, Lokshtanov, Penninkx, Saurabh, Thilikos

(Meta) Kernelization. FOCS 2009: 629-638 Used before in the

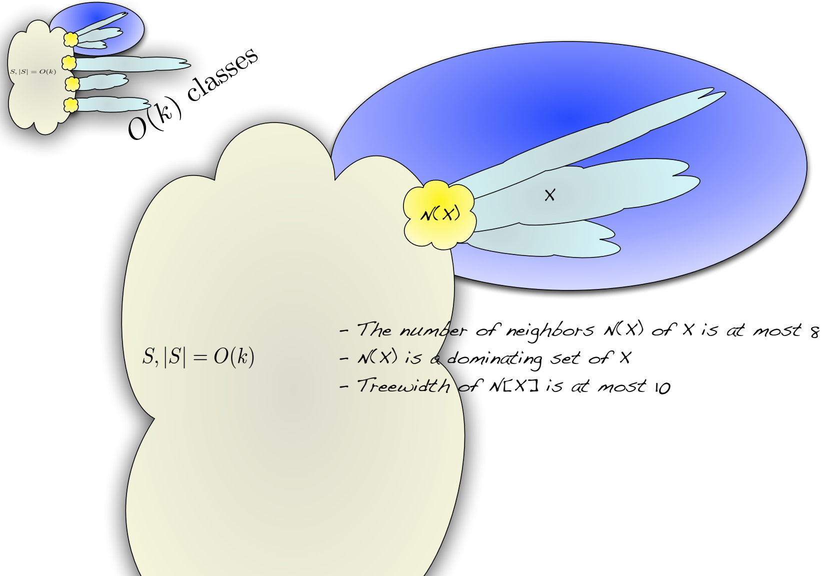
development of algorithms on graphs of bounded treewidth.

How protrusions work for parameterized problem Π

If the size of protrusion X is larger than some constant x (depending only on Π), it is possible to replace X by a protrusion X' of size $x' < x$ such that the solution for Π remains the "same" on the new graph.



In the dominating set example protrusions were the sets X formed by different classes of neighborhoods in S.



Protrusion replacer

A protrusion replacer is an algorithm (well, a sequence of algorithms, one for each r) that in polynomial time reduces each protrusion to the size $f(r)$.

Protrusion replacer

A protrusion replacer is an algorithm (well, a sequence of algorithms, one for each r) that in polynomial time reduces each protrusion to the size $f(r)$.

For DS we use $r \leq 8$.

Theorem

Every separable CMSO-Optimization problem has a protrusion replacer.

Putting things together

Theorem

Every bidimensional separable CMSO-Optimization problem has a linear kernel on planar graphs.

Putting things together

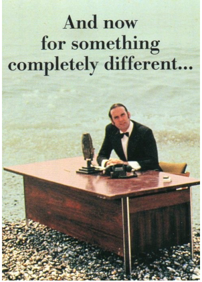
Theorem

Every bidimensional separable CMSO-Optimization problem has a linear kernel on planar graphs.

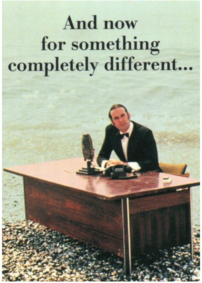
Where did we use planarity?

CMSO Bidimensional separable problems

Dominating Set, r-Dominating Set, Vertex Cover, Connected r-Dominating Set, Connected Vertex Cover, Vertex-Minor-Covering, Minimum Maximal Matching, Vertex-Subgraph-Covering, Clique-Transversal, Almost-Outerplanar, Feedback Vertex Set, Cycle Domination, Edge Dominating Set, Independent Set, Induced d-Degree Subgraph, r-Scattered Set, Induced Matching, Triangle Packing, Cycle Packing ...



- School on Recent Advances in Parameterized Complexity will be held on December 3-7, 2017, Tel Aviv, Israel.



- School on Recent Advances in Parameterized Complexity will be held on December 3-7, 2017, Tel Aviv, Israel.
- PhD position in Bergen (send me email)