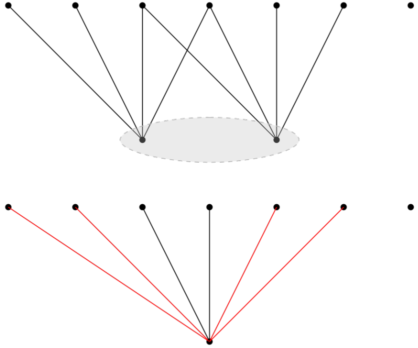
Computing Twin-Width via Treedepth and Vertex Integrity

Robert Ganian, Mathis Rocton

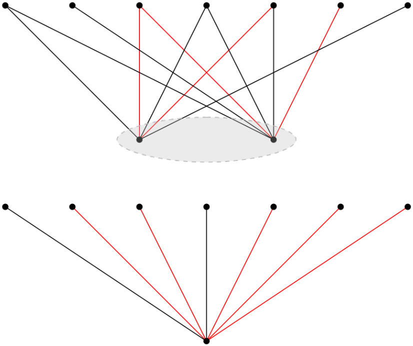
TU Wien

March 11, 2026

Contracting Vertices



Contracting Vertices



Contraction Sequences

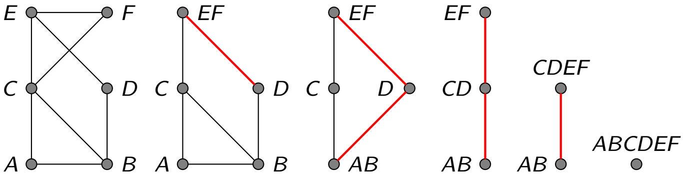


Figure by Jakub Balabán.

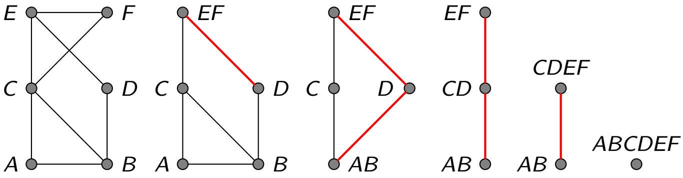


Figure by Jakub Balabán.

The *twin-width* (tww) of a graph G is the minimum width of a contraction sequence, over all valid contraction sequences from G to K_1.

Using Twin-Width to Solve FO

Theorem (Bonnet, Kim, Thomassé, Watrigant; 2020)

Provided a ***contraction sequence*** *of G of width d, evaluating a formula φ expressible in* ***First Order Logic (FO)*** *on G can be done in time $f(d, |\varphi|) \cdot |V(G)|$ for a computable function f.*

Using Twin-Width to Solve FO

Theorem (Bonnet, Kim, Thomassé, Watrigant; 2020)

Provided a ***contraction sequence*** *of G of width d, evaluating a formula φ expressible in* ***First Order Logic (FO)*** *on G can be done in time $f(d, |\varphi|) \cdot |V(G)|$ for a computable function f.*

To this date, we do **not** have any efficient way of computing contraction sequences of (quasi) optimal width!

Using Twin-Width to Solve FO

Theorem (Bonnet, Kim, Thomassé, Watrigant; 2020)

Provided a ***contraction sequence*** *of G of width d, evaluating a formula φ expressible in* ***First Order Logic (FO)*** *on G can be done in time $f(d, |\varphi|) \cdot |V(G)|$ for a computable function f.*

To this date, we do **not** have any efficient way of computing contraction sequences of (quasi) optimal width!

Theorem (Bergé, Bonnet, Déprés; 2022)

Deciding whether the twin-width of a graph is at most 4 is NP-complete.

Computing Twin-Width

Is there an algorithm that given an n-vertex graph G and $k \in \mathbb{N}$, runs in time $f(k) \cdot n^{\mathcal{O}(1)}$ and either correctly reports that $\text{tww}(G) \geq k$ or outputs a contr. sequence of width at most $g(k)$?

- At this moment wide open!

Reminder: An algorithm is FPT parameterized by k if it runs in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable $f(.)$.

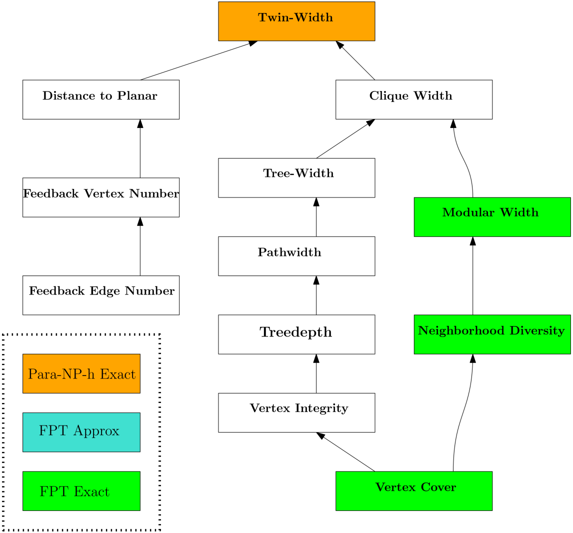
Computing Twin-Width

Is there an algorithm that given an n-vertex graph G and $k \in \mathbb{N}$, runs in time $f(k) \cdot n^{\mathcal{O}(1)}$ and either correctly reports that $\text{tww}(G) \geq k$ or outputs a contr. sequence of width at most $g(k)$?

- At this moment wide open!
- What about using more restrictive parameters?
- Then maybe even *exact* FPT algorithms would be possible!

Reminder: An algorithm is FPT parameterized by k if it runs in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable $f(.)$.

State of the Art



State of the Art

Theorem (+1-approximation for FEN)

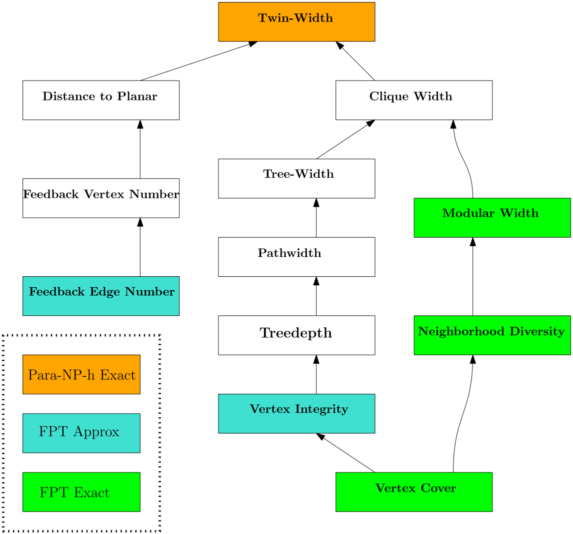
It is FPT to compute a contraction sequence for G of width at most $\text{tww}(G) + 1$*, parameterized by the Feedback Edge Number.*

Theorem (2-approximation for VI)

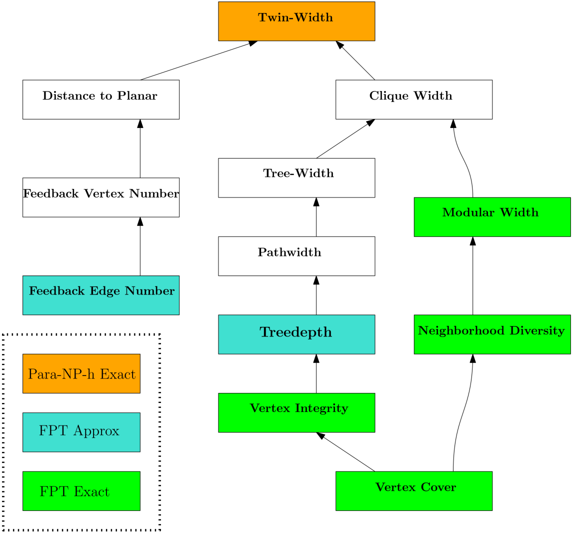
It is FPT to compute a contraction sequence for G of width at most $2 \cdot \text{tww}(G)$*, parameterized by the Vertex Integrity.*

Balabán, Ganian, R.; 2025

State of the Art



State of the Art



Improvements in Two Directions

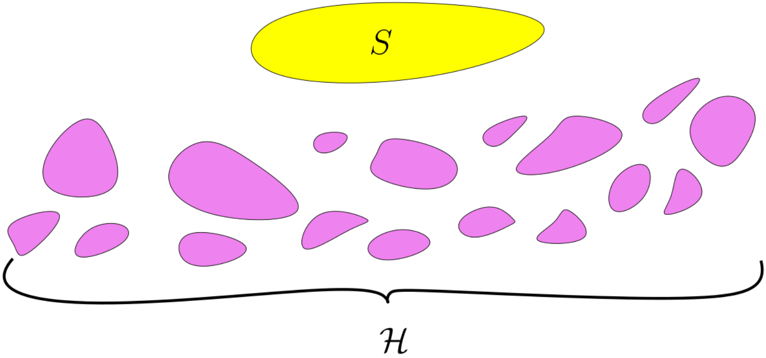
Theorem 1

Computing an **optimal** contraction sequence is FPT parameterized by the vertex integrity.

Theorem 2

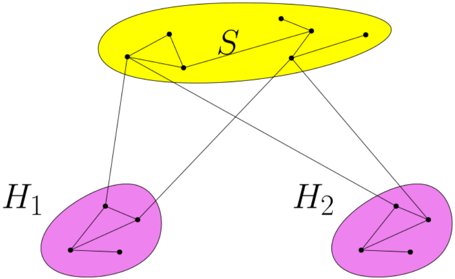
For some computable function q, computing a contraction sequence of width at most $q(\text{tww})$ is FPT parameterized by treedepth.

Vertex Integrity

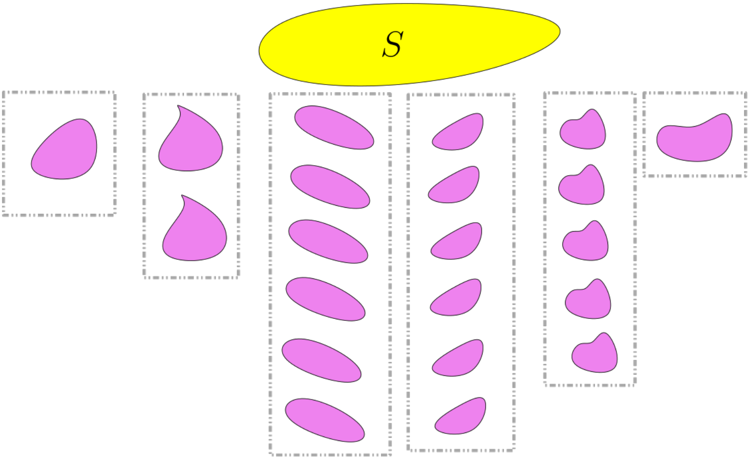


Vertex Integrity

Vertex Integrity

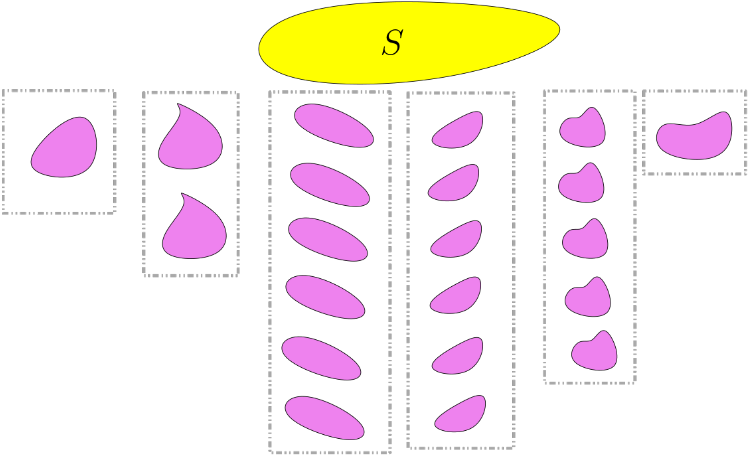


Classical Pruning



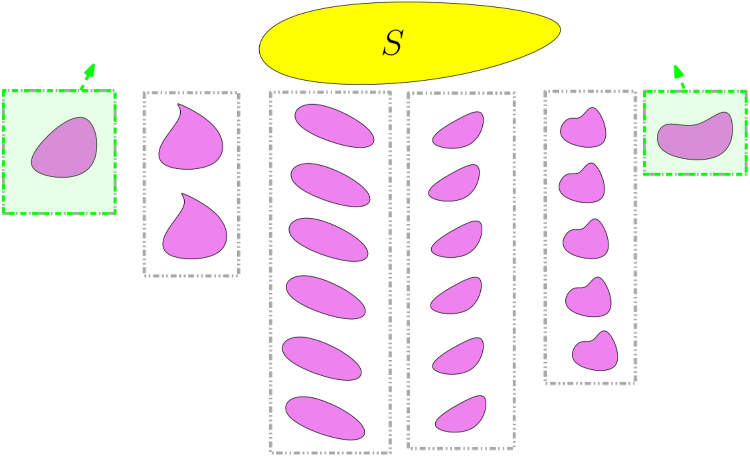
Technique introduced for minimum-page stack and queue layouts

[Depian, Fink, Ganian, Surianarayanan 25']



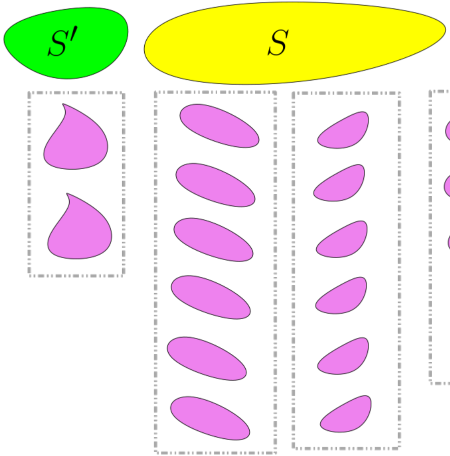
Technique introduced for minimum-page stack and queue layouts

[Depian, Fink, Ganian, Surianarayanan 25']



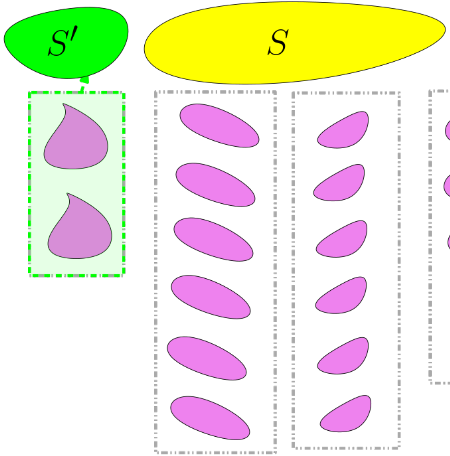
Technique introduced for minimum-page stack and queue layouts

[Depian, Fink, Ganian, Surianarayanan 25']



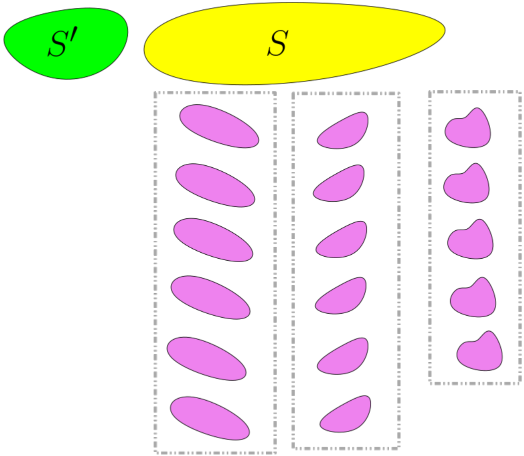
Technique introduced for minimum-page stack and queue layouts

[Depian, Fink, Ganian, Surianarayanan 25']

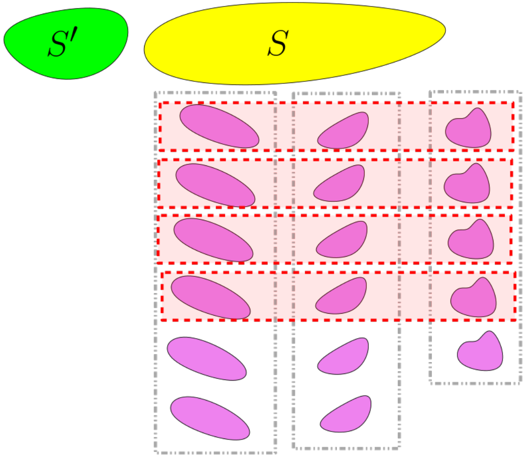


Technique introduced for minimum-page stack and queue layouts

[Depian, Fink, Ganian, Surianarayanan 25']

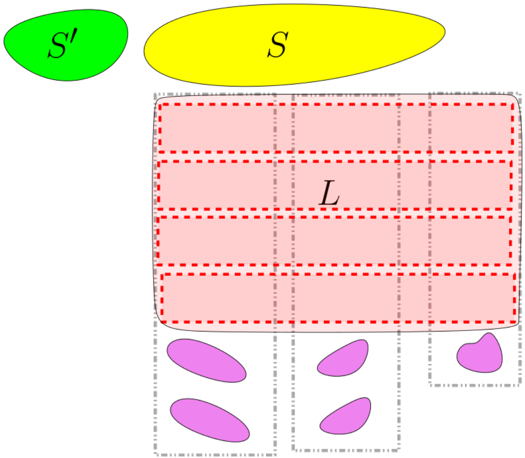


Technique introduced for minimum-page stack and queue layouts
[Depian, Fink, Ganian, Surianarayanan 25']



Technique introduced for minimum-page stack and queue layouts

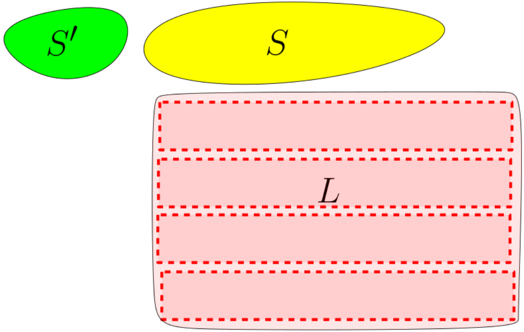
[Depian, Fink, Ganian, Surianarayanan 25']



Ramsey Pruning

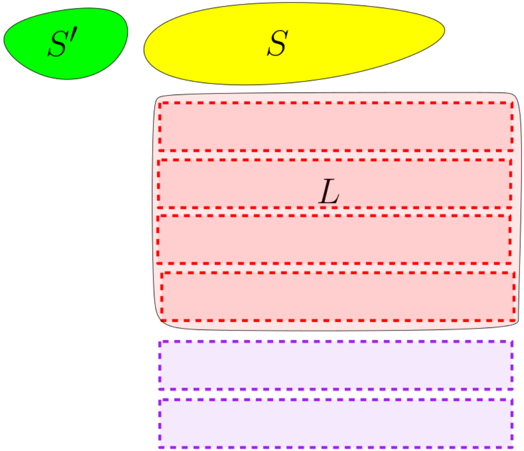
Technique introduced for minimum-page stack and queue layouts

[Depian, Fink, Ganian, Surianarayanan 25']



Technique introduced for minimum-page stack and queue layouts

[Depian, Fink, Ganian, Surianarayanan 25']



Hypergraph Ramsey

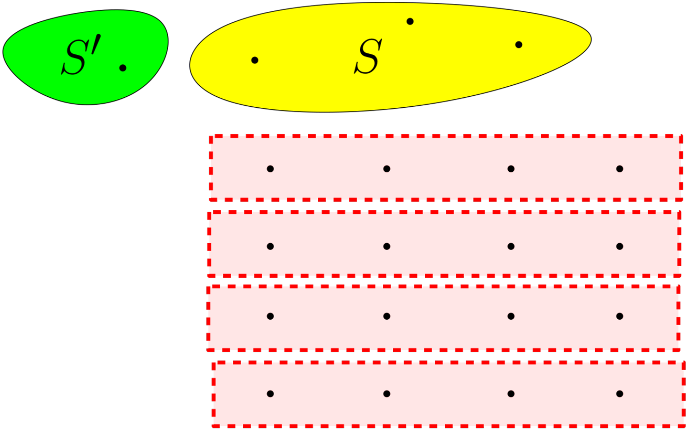
- We want to capture the trace of a potential solution on the *red blocks* . . .
- . . . but also the interaction between blocks . . .
- . . . and the order of these interactions.

Hypergraph Ramsey

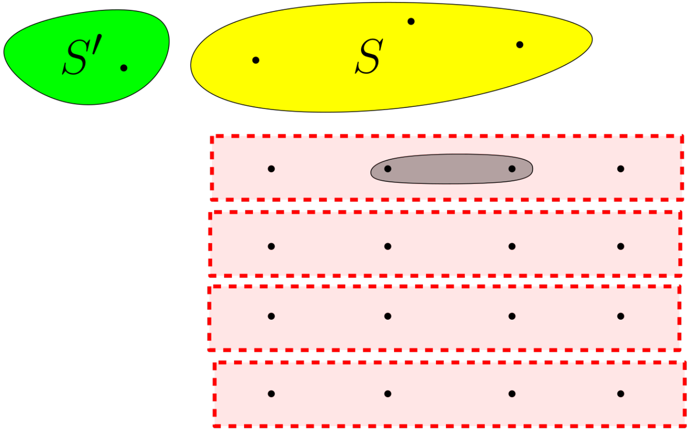
- We want to capture the trace of a potential solution on the *red blocks* . . .
- . . . but also the interaction between blocks . . .
- . . . and the order of these interactions.

$\rightarrow$ We label **triplets** (L_1, L_2, L_3) with the restriction of the solution to $G[S \cup S' \cup L_1 \cup L_2 \cup L_3]$, and Hypergraph Ramsey gives us a family $\mathcal{L}' \subset L$ in which all triplets have the same label.

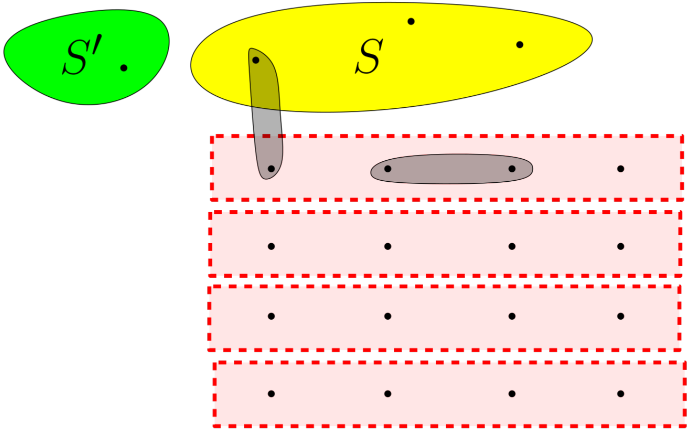
Tame Solution



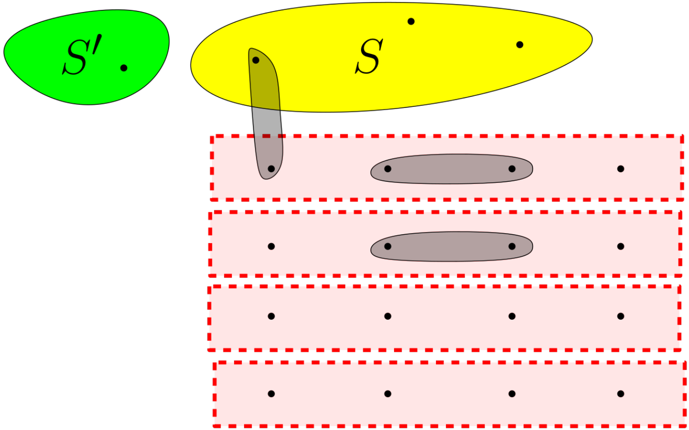
Tame Solution



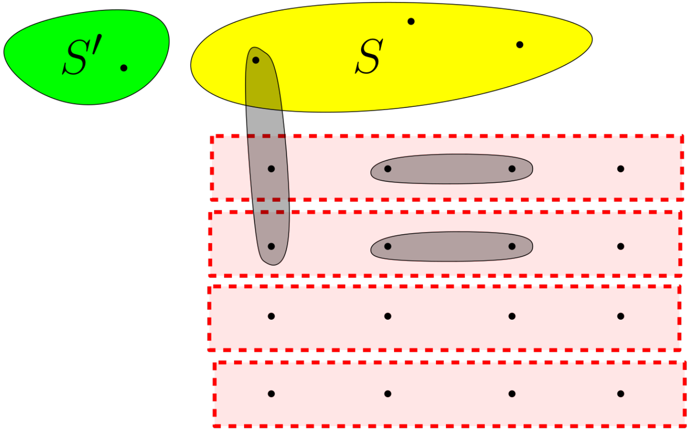
Tame Solution



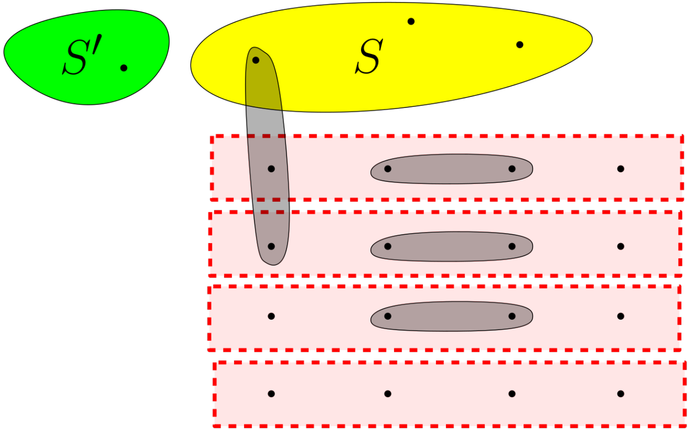
Tame Solution



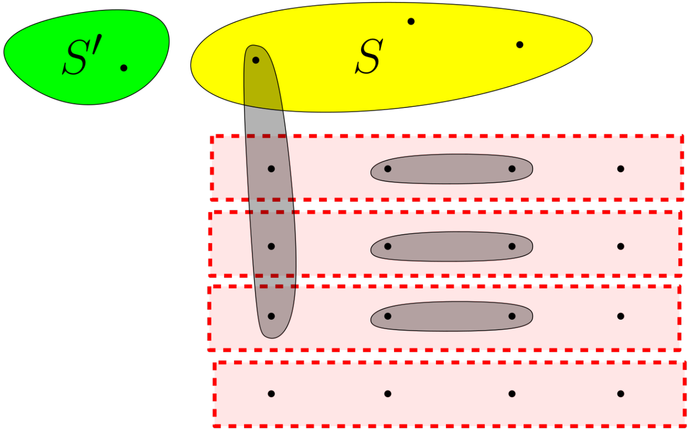
Tame Solution



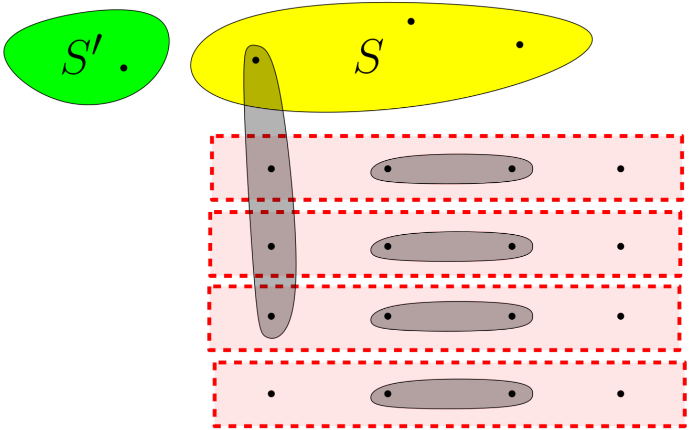
Tame Solution



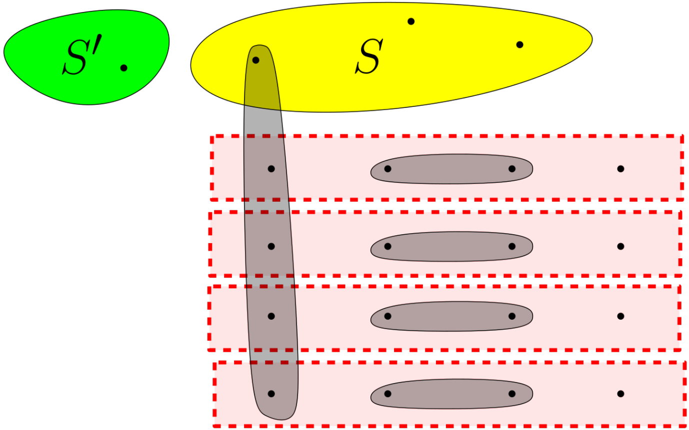
Tame Solution



Tame Solution



Tame Solution



The Cost of Ramsey

The size of the kernel is:

$$2 \uparrow\uparrow ((p \cdot 2^{2p^2} \cdot 6 + 6)^p + 10)$$

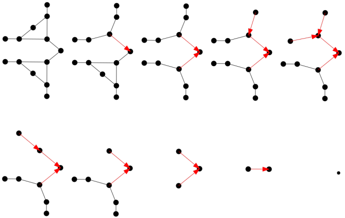
...definitely not tight

From Vertex Integrity to Treedepth

- Vertex Integrity is too restrictive: Treedepth is more general.
- No known equivalent to the Ramsey Pruning technique.
- Classical pruning approach for Vertex Integrity (2-approx.) can be adapted for Treedepth ...

  ... but yields a 2^{td}-approximation.
- **Workaround:** the *oriented* twin-width (otww).

Making Twin-Width Oriented



- Refines the error edges with orientation
- Only red out-degree is taken into account
- Functionally equivalent to Twin-Width

→ Near-optimal contractions for otww yield near-optimal contractions for tww in FPT time!

(combining results from Twin-Width I, IV and VI)

Open directions

- Ramsey Pruning for other problems
- Ramsey Pruning generalization to Treedepth
- Is oriented twin-width the "right" parameter to crack twin-width approximability?
- Approximating twin-width parameterized by pathwidth/treewidth

Open directions

- Ramsey Pruning for other problems
- Ramsey Pruning generalization to Treedepth
- Is oriented twin-width the "right" parameter to crack twin-width approximability?
- Approximating twin-width parameterized by pathwidth/treewidth

Thank you for your attention!
Questions?