Hitting forbidden subgraphs in graphs of bounded treewidth

Marek Cygan, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk

MFCS 2014, Budapest,
August 25^{th}, 2014

The problem

H-Subgraph Hitting

Input: A graph G and an integer k

Question: Does there exist a subset of vertices X with $|X| \leq k$ such that $G - X$ does not contain H as a subgraph?

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.
  - Parameters k or $|H|$.

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.
  - Parameters k or $|H|$.
- In this work we consider parameterization by the *treewidth*.

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.
  - Parameters k or $|H|$.
- In this work we consider parameterization by the *treewidth*.
  - A graph has treewidth t if it resembles a tree with the trunk having width roughly t.

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.
  - Parameters k or $|H|$.
- In this work we consider parameterization by the *treewidth*.
  - A graph has treewidth t if it resembles a tree with the trunk having width roughly t.
  - **Setting**: G is given together with a tree decomposition of width t.

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.
  - Parameters k or $|H|$.
- In this work we consider parameterization by the *treewidth*.
  - A graph has treewidth t if it resembles a tree with the trunk having width roughly t.
  - **Setting**: G is given together with a tree decomposition of width t.
  - H is considered fixed and constant-size.

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.
  - Parameters k or $|H|$.
- In this work we consider parameterization by the *treewidth*.
  - A graph has treewidth t if it resembles a tree with the trunk having width roughly t.
  - **Setting**: G is given together with a tree decomposition of width t.
  - H is considered fixed and constant-size.
  - The width of the decomposition t is the parameter.

The setting

- A fundamental graph modification problem, studied in parameterized complexity w.r.t. different parameters.
  - Parameters k or $|H|$.
- In this work we consider parameterization by the *treewidth*.
  - A graph has treewidth t if it resembles a tree with the trunk having width roughly t.
  - **Setting**: G is given together with a tree decomposition of width t.
  - H is considered fixed and constant-size.
  - The width of the decomposition t is the parameter.
  - k can be arbitrarily large; we do not consider it as a parameter.

Preliminary observations

- An algorithm with running time $f(|H|, t) \cdot n$ follows from the Courcelle's theorem.

Preliminary observations

- An algorithm with running time $f(|H|, t) \cdot n$ follows from the Courcelle's theorem.
- It is not hard to make an algorithm with running time

$$2^{f(|H|) \cdot t^{\mathcal{O}(|H|)}} \cdot n^{\mathcal{O}(1)}.$$

Preliminary observations

- An algorithm with running time $f(|H|, t) \cdot n$ follows from the Courcelle's theorem.
- It is not hard to make an algorithm with running time

$$2^{f(|H|)\cdot t^{\mathcal{O}(|H|)}} \cdot n^{\mathcal{O}(1)}.$$

- There are technical caveats.

Preliminary observations

- An algorithm with running time $f(|H|, t) \cdot n$ follows from the Courcelle's theorem.
- It is not hard to make an algorithm with running time

$$2^{f(|H|)\cdot t^{\mathcal{O}(|H|)}} \cdot n^{\mathcal{O}(1)}.$$

- There are technical caveats.
- **Our question**: what is the right exponent at the $\mathrm{poly}(t)$ factor?

Hitting cycles

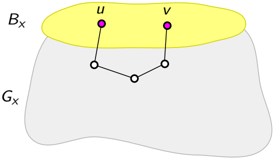
- Consider the problem of, say, hitting all the C_ℓ-s in the graph.

Hitting cycles

- Consider the problem of, say, hitting all the C_ℓ-s in the graph.
- (P., MFCS 2011) For $\ell \geq 5$, there is no algorithm with running time $2^{o(t^2)} \cdot n^{\mathcal{O}(1)}$ for this problem, unless ETH fails.

Hitting cycles

- Consider the problem of, say, hitting all the C_ℓ-s in the graph.
- (P., MFCS 2011) For $\ell \geq 5$, there is no algorithm with running time $2^{o(t^2)} \cdot n^{\mathcal{O}(1)}$ for this problem, unless ETH fails.
- *"[...] these problems admit a simple $2^{\mathcal{O}(t^2)} n^{\mathcal{O}(1)}$ dynamic programming algorithm, where t is the width of a given tree decomposition. In the state, one remembers for every pair of vertices of bag B_x, whether in G_x they can be connected via paths of length $1, 2, \ldots, \ell - 1$ disjoint with the solution."*



Hitting cycles

- More precisely, we consider a family $\mathcal{F}$ of *folios* ϕ:

Hitting cycles

- More precisely, we consider a family $\mathcal{F}$ of *folios* ϕ:
  - Folio ϕ says, for every pair $u, v \in B_x$ and every $p = 1, 2, \ldots, \ell - 1$, whether there should be a path of length p between u and v below the bag.

Hitting cycles

- More precisely, we consider a family $\mathcal{F}$ of *folios* ϕ:
  - Folio ϕ says, for every pair $u, v \in B_x$ and every $p = 1, 2, \ldots, \ell - 1$, whether there should be a path of length p between u and v below the bag.
  - $|\mathcal{F}| = \mathcal{O}(t^2)$.

Hitting cycles

- More precisely, we consider a family $\mathcal{F}$ of *folios* ϕ:
  - Folio ϕ says, for every pair $u, v \in B_x$ and every $p = 1, 2, \ldots, \ell - 1$, whether there should be a path of length p between u and v below the bag.
  - $|\mathcal{F}| = \mathcal{O}(t^2)$.
- The DP table $T[x, \phi]$: for every $\phi \in \mathcal{F}$, the minimum number of vertices to remove to achieve folio ϕ.

Hitting cycles

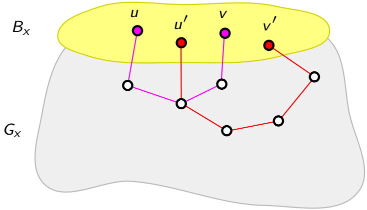
- More precisely, we consider a family $\mathcal{F}$ of *folios* ϕ:
  - Folio ϕ says, for every pair $u, v \in B_x$ and every $p = 1, 2, \ldots, \ell - 1$, whether there should be a path of length p between u and v below the bag.
  - $|\mathcal{F}| = \mathcal{O}(t^2)$.
- The DP table $T[x, \phi]$: for every $\phi \in \mathcal{F}$, the minimum number of vertices to remove to achieve folio ϕ.
- In the updates, the information about the folios should be sufficient to ensure that no C_ℓ is created.

Hitting cycles

- Is it really?

Hitting cycles

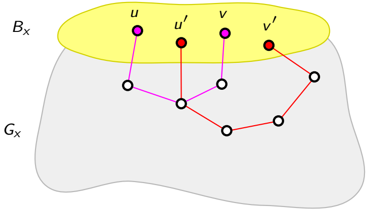
- Is it really?



- We also need to store information about intersecting paths.

Hitting cycles

- Is it really?



- We also need to store information about intersecting paths.
- Starts to be a mess.

Results, informally

- **Question**: What is optimum exponent at t in the $\mathrm{poly}(t)$ factor? How does it depend on a general (connected) H?

Results, informally

- **Question**: What is optimum exponent at t in the $\mathrm{poly}(t)$ factor? How does it depend on a general (connected) H?
- **Our results**:

Results, informally

- **Question**: What is optimum exponent at t in the $\mathrm{poly}(t)$ factor? How does it depend on a general (connected) H?
- **Our results**:
  - For *colorful variant*, we get tight bounds on what this exponent should be.

Results, informally

- **Question**: What is optimum exponent at t in the $\mathrm{poly}(t)$ factor? How does it depend on a general (connected) H?
- **Our results**:
  - For *colorful variant*, we get tight bounds on what this exponent should be.
  - For standard variant, we get an upper bound that is weaker than for the colorful variant.

Results, informally

- **Question**: What is optimum exponent at t in the $\mathrm{poly}(t)$ factor? How does it depend on a general (connected) H?
- **Our results**:
  - For *colorful variant*, we get tight bounds on what this exponent should be.
  - For standard variant, we get an upper bound that is weaker than for the colorful variant.
    - For hitting cycles: $2^{\mathcal{O}(t^2 \log t)} \cdot n$ algorithm.

Results, informally

- **Question**: What is optimum exponent at t in the $\mathrm{poly}(t)$ factor? How does it depend on a general (connected) H?
- **Our results**:
  - For *colorful variant*, we get tight bounds on what this exponent should be.
  - For standard variant, we get an upper bound that is weaker than for the colorful variant.
    - For hitting cycles: $2^{\mathcal{O}(t^2 \log t)} \cdot n$ algorithm.
  - However, there are cases when the complexity of the standard variant is provably *lower* than the colorful one.

Results, informally

- **Question**: What is optimum exponent at t in the $\mathrm{poly}(t)$ factor? How does it depend on a general (connected) H?
- **Our results**:
  - For *colorful variant*, we get tight bounds on what this exponent should be.
  - For standard variant, we get an upper bound that is weaker than for the colorful variant.
    - For hitting cycles: $2^{\mathcal{O}(t^2 \log t)} \cdot n$ algorithm.
  - However, there are cases when the complexity of the standard variant is provably *lower* than the colorful one.
  - The exponent is governed by sizes of minimal separators in H.

The colorful problem

Colorful H-Subgraph Hitting

Input: A graph G with vertices partitioned into classes $\{V_u \colon u \in V(H)\}$ and an integer k

Question: Is there a subset of vertices X with $|X| \leq k$ such that $G - X$ does not contain a colorful H-subgraph?

The colorful problem

COLORFUL H-SUBGRAPH HITTING

Input: A graph G with vertices partitioned into classes $\{V_u : u \in V(H)\}$ and an integer k

Question: Is there a subset of vertices X with $|X| \leq k$ such that $G - X$ does not contain a colorful H-subgraph?

- Colorful H-subgraph: $\phi(u) \in V_u$, for each $u \in V(H)$.

The colorful problem

Colorful H-Subgraph Hitting

Input: A graph G with vertices partitioned into classes $\{V_u \colon u \in V(H)\}$ and an integer k

Question: Is there a subset of vertices X with $|X| \leq k$ such that $G - X$ does not contain a colorful H-subgraph?

- Colorful H-subgraph: $\phi(u) \in V_u$, for each $u \in V(H)$.
- **Remark**: no problem with intersecting parts in the DP!

Combinatorial invariants

$$\mu(G) = \max\{|S| : S \text{ is a minimal separator}\}.$$

Combinatorial invariants

$$\mu(G) = \max\{|S| : S \text{ is a minimal separator}\}.$$

- $S \subseteq V(G)$ is a minimal separator if $G - S$ contains at least two connected components that are full towards S.

Combinatorial invariants

$$\mu(G) = \max\{|S| : S \text{ is a minimal separator}\}.$$

- $S \subseteq V(G)$ is a minimal separator if $G - S$ contains at least two connected components that are full towards S.
- A component C is full to S if $N_G(V(C)) = S$.

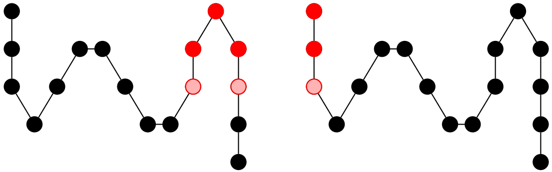
Combinatorial invariants

$$\mu(G) = \max\{|S| : S \text{ is a minimal separator}\}.$$

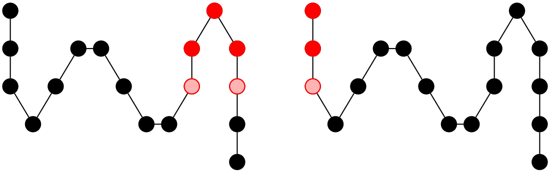
- $S \subseteq V(G)$ is a minimal separator if $G - S$ contains at least two connected components that are full towards S.
- A component C is full to S if $N_G(V(C)) = S$.

$$\mu^\star(G) = \max\{|N(X)| : G[X] \text{ is connected and } N[X] \neq V(H)\}.$$

... on a picture

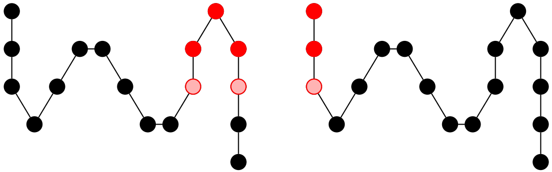


... on a picture



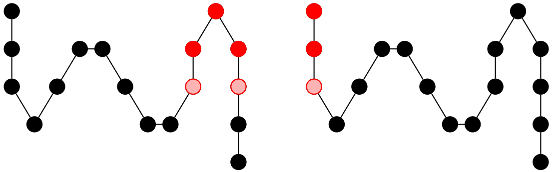
- $\mu(P_\ell) = 1$, $\mu^\star(P_\ell) = 2$.

... on a picture



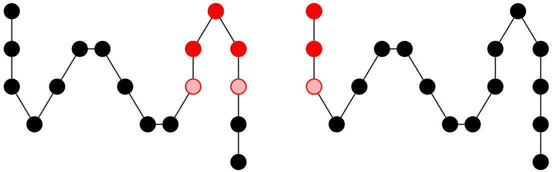
- $\mu(P_\ell) = 1$, $\mu^\star(P_\ell) = 2$.
- $\mu(C_\ell) = 2$, $\mu^\star(C_\ell) = 2$.

... on a picture



- $\mu(P_\ell) = 1$, $\mu^\star(P_\ell) = 2$.
- $\mu(C_\ell) = 2$, $\mu^\star(C_\ell) = 2$.
- $\mu(H) \leq \mu^\star(H)$.

... on a picture



- $\mu(P_\ell) = 1$, $\mu^\star(P_\ell) = 2$.
- $\mu(C_\ell) = 2$, $\mu^\star(C_\ell) = 2$.
- $\mu(H) \leq \mu^\star(H)$.
- **Note**: μ and $\mu^\star$ are undefined for a clique.

Results, formally

Theorem 1

For connected H not being a clique, colorful variant can be solved in time $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.

Results, formally

Theorem 1

For connected H not being a clique, colorful variant can be solved in time $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.

Theorem 2

For connected H not being a clique nor a path, colorful variant cannot be solved in time $2^{o(t^{\mu(H)})} \cdot n^{\mathcal{O}(1)}$ unless ETH fails.

Results, formally

Theorem 1

For connected H not being a clique, colorful variant can be solved in time $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.

Theorem 2

For connected H not being a clique nor a path, colorful variant cannot be solved in time $2^{o(t^{\mu(H)})} \cdot n^{\mathcal{O}(1)}$ unless ETH fails.

- **Surprise**: hitting a colorful path is polynomial-time solvable.

Results, formally

Theorem 1

For connected H not being a clique, colorful variant can be solved in time $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.

Theorem 2

For connected H not being a clique nor a path, colorful variant cannot be solved in time $2^{o(t^{\mu(H)})} \cdot n^{\mathcal{O}(1)}$ unless ETH fails.

- **Surprise**: hitting a colorful path is polynomial-time solvable.

Theorem 3

For connected H not being a clique, standard variant can be solved in time $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)} \cdot n$.

Theorem 1: intuition

- Colors $\Rightarrow$ No problems with intersections

Theorem 1: intuition

- Colors $\Rightarrow$ No problems with intersections
- Standard DP gives $2^{\mathcal{O}(t^{\mu^\star(H)})} \cdot n$, but we want $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.

Theorem 1: intuition

- Colors $\Rightarrow$ No problems with intersections
- Standard DP gives $2^{\mathcal{O}(t^{\mu^\star(H)})} \cdot n$, but we want $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.
  - **Concrete case**: a $2^{\mathcal{O}(t)} \cdot n$ DP for hitting a colorful path.

Theorem 1: intuition

- Colors $\Rightarrow$ No problems with intersections
- Standard DP gives $2^{\mathcal{O}(t^{\mu^\star(H)})} \cdot n$, but we want $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.
  - **Concrete case**: a $2^{\mathcal{O}(t)} \cdot n$ DP for hitting a colorful path.
- To get a $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$-time algorithm, we use *prediction*.

Theorem 1: intuition

- Colors $\Rightarrow$ No problems with intersections
- Standard DP gives $2^{\mathcal{O}(t^{\mu^\star(H)})} \cdot n$, but we want $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$.
  - **Concrete case**: a $2^{\mathcal{O}(t)} \cdot n$ DP for hitting a colorful path.
- To get a $2^{\mathcal{O}(t^{\mu(H)})} \cdot n$-time algorithm, we use *prediction*.
- Find the smallest solution in the subgraph that is consistent with a predicted final behaviour in the bag for the whole graph.

Theorem 2: intuition

- Generalize the construction for hitting cycles from the MFCS 2011 paper.

Theorem 2: intuition

- Generalize the construction for hitting cycles from the MFCS 2011 paper.
- Under ETH, there is no $2^{o(n+m)}$ algorithm for $3\mathrm{SAT}$, where n is the number of variables and m is the number of clauses.

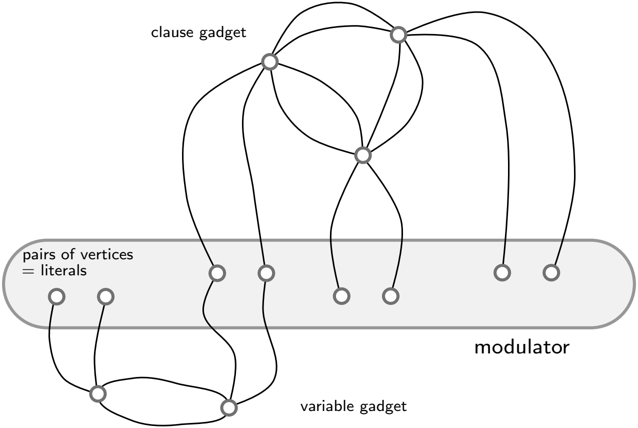
Theorem 2: intuition

- Generalize the construction for hitting cycles from the MFCS 2011 paper.
- Under ETH, there is no $2^{o(n+m)}$ algorithm for $\mathrm{3SAT}$, where n is the number of variables and m is the number of clauses.
- Start with an instance φ of $\mathrm{3SAT}$, output an equivalent instance of Colorful H-Subgraph Hitting with treewidth $\mathcal{O}(\sqrt{n+m})$.

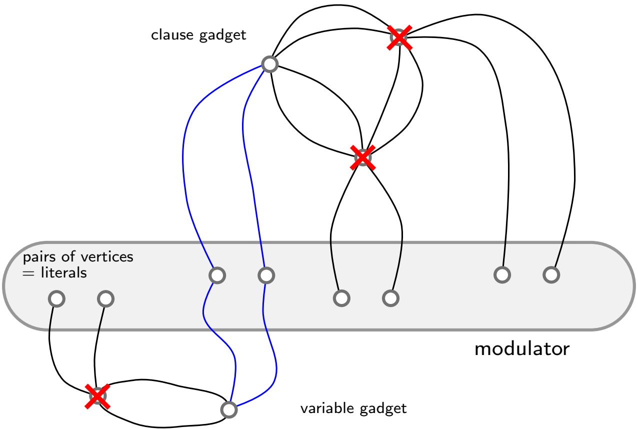
Theorem 2: intuition

- Generalize the construction for hitting cycles from the MFCS 2011 paper.
- Under ETH, there is no $2^{o(n+m)}$ algorithm for $\mathrm{3SAT}$, where n is the number of variables and m is the number of clauses.
- Start with an instance φ of $\mathrm{3SAT}$, output an equivalent instance of Colorful H-Subgraph Hitting with treewidth $\mathcal{O}(\sqrt{n+m})$.
- The total number of literals is at most $3m$.

Theorem 2: intuition



Theorem 2: intuition



Theorem 2: intuition

- In the proof we use the fact that a cycle has two sides,

Theorem 2: intuition

- In the proof we use the fact that a cycle has two sides,
- Replace two sides of the cycle with two sides of the separator.

Theorem 2: intuition

- In the proof we use the fact that a cycle has two sides,
- Replace two sides of the cycle with two sides of the separator.
- Embed the space of literals into $\mu(H)$-sized subsets of the modulator.

Theorem 2: intuition

- In the proof we use the fact that a cycle has two sides,
- Replace two sides of the cycle with two sides of the separator.
- Embed the space of literals into $\mu(H)$-sized subsets of the modulator.
- Colors help us not to create unwanted copies.

Theorem 3: intuition

- **Main idea**: the witness graph.

Theorem 3: intuition

- **Main idea**: the witness graph.
- Suppose G_x is the subgraph below the bag B_x.

Theorem 3: intuition

- **Main idea**: the witness graph.
- Suppose G_x is the subgraph below the bag B_x.
- For some $X \subseteq V(G_x)$, we say that a graph P is a *witness graph* for $G_x - X$ if:

Theorem 3: intuition

- **Main idea**: the witness graph.
- Suppose G_x is the subgraph below the bag B_x.
- For some $X \subseteq V(G_x)$, we say that a graph P is a *witness graph* for $G_x - X$ if:
  - $P[B_x \cap V(P)] = G[B_x \setminus X]$;

Theorem 3: intuition

- **Main idea**: the witness graph.
- Suppose G_x is the subgraph below the bag B_x.
- For some $X \subseteq V(G_x)$, we say that a graph P is a *witness graph* for $G_x - X$ if:
  - $P[B_x \cap V(P)] = G[B_x \setminus X]$;
  - for any possible "outhanging" subgraph H' of H, there is a model of H' in $G_x - X$ hanging below B_x iff there is one in P.

Theorem 3: intuition

- **Main idea**: the witness graph.
- Suppose G_x is the subgraph below the bag B_x.
- For some $X \subseteq V(G_x)$, we say that a graph P is a *witness graph* for $G_x - X$ if:
  - $P[B_x \cap V(P)] = G[B_x \setminus X]$;
  - for any possible "outhanging" subgraph H' of H, there is a model of H' in $G_x - X$ hanging below B_x iff there is one in P.
  - **Note**: H' can be disconnected in this definition.

Theorem 3: intuition

- **Main ingredient**: there is always a witness graph that has $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.

Theorem 3: intuition

- **Main ingredient**: there is always a witness graph that has $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- There are $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)}$ graphs with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.

Theorem 3: intuition

- **Main ingredient**: there is always a witness graph that has $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- There are $2^{\mathcal{O}(t^{\mu^\star(H)}\log t)}$ graphs with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- **DP states**: graphs with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.

Theorem 3: intuition

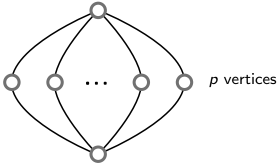
- **Main ingredient**: there is always a witness graph that has $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- There are $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)}$ graphs with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- **DP states**: graphs with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- **DP function**: $T[x, P]$ is the minimum number of vertices needed to delete below the bag in order to obtain a subgraph equivalent to P.

Theorem 3: intuition

- **Main ingredient**: there is always a witness graph that has $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- There are $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)}$ graphs with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- **DP states**: graphs with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges.
- **DP function**: $T[x, P]$ is the minimum number of vertices needed to delete below the bag in order to obtain a subgraph equivalent to P.
- Many technicalities omitted.

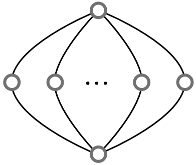
Difference between variants

- Consider the *pumpkin graph* θ_p.



Difference between variants

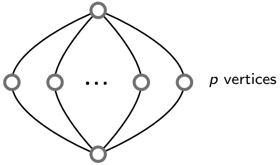
- Consider the *pumpkin graph* θ_p.



- Colorful variant: no $2^{o(t^p)} \cdot n^{\mathcal{O}(1)}$ algorithm.

Difference between variants

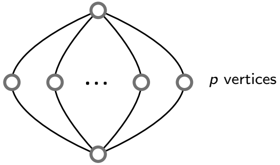
- Consider the *pumpkin graph* θ_p.



- Colorful variant: no $2^{o(t^p)} \cdot n^{\mathcal{O}(1)}$ algorithm.
- Standard variant: a $2^{\mathcal{O}(t^2 \log t)} \cdot n$ algorithm.

Difference between variants

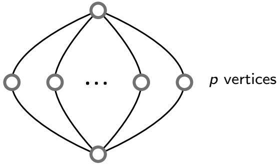
- Consider the *pumpkin graph* θ_p.



- Colorful variant: no $2^{o(t^p)} \cdot n^{\mathcal{O}(1)}$ algorithm.
- Standard variant: a $2^{\mathcal{O}(t^2 \log t)} \cdot n$ algorithm.
- If one attaches triangles to the top and bottom, then a $2^{o(t^p)} \cdot n^{\mathcal{O}(1)}$ lower bound in the standard variant.

Difference between variants

- Consider the *pumpkin graph* θ_p.



- Colorful variant: no $2^{o(t^p)} \cdot n^{\mathcal{O}(1)}$ algorithm.
- Standard variant: a $2^{\mathcal{O}(t^2 \log t)} \cdot n$ algorithm.
- If one attaches triangles to the top and bottom, then a $2^{o(t^p)} \cdot n^{\mathcal{O}(1)}$ lower bound in the standard variant.
- Seems like the answer depends on the symmetries of H.

Open problems

- Obtain a $2^{\mathcal{O}(t^{\mu(H)} \log t)} \cdot n^{\mathcal{O}(1)}$ algorithm for the standard variant as well.

Open problems

- Obtain a $2^{\mathcal{O}(t^{\mu(H)} \log t)} \cdot n^{\mathcal{O}(1)}$ algorithm for the standard variant as well.
  - It is possible for paths.

Open problems

- Obtain a $2^{\mathcal{O}(t^{\mu(H)} \log t)} \cdot n^{\mathcal{O}(1)}$ algorithm for the standard variant as well.
  - It is possible for paths.
- Minor hitting?

Open problems

- Obtain a $2^{\mathcal{O}(t^{\mu(H)}\log t)} \cdot n^{\mathcal{O}(1)}$ algorithm for the standard variant as well.
  - It is possible for paths.
- Minor hitting?
- **Thanks for attention!**