Structural sparsity and parameterized algorithms

Michał Pilipczuk

Institute of Informatics, University of Warsaw, Poland

Recent Advances in Parameterized Complexity

Tel Aviv, December 7^{th}, 2017

What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?

What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?
  - **Positive ex**: Bounded max degree, planar, bounded treewidth, ...

What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?
  - **Positive ex**: Bounded max degree, planar, bounded treewidth, ...
  - **Negative ex**: Cliques, bicliques, ...

What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?
  - **Positive ex**: Bounded max degree, planar, bounded treewidth, ...
  - **Negative ex**: Cliques, bicliques, ...
- **Attempt 1.** Edge density bounded by a constant:

$$\mathrm{density}(G) := \frac{|E(G)|}{|V(G)|} \leqslant c.$$

What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?
  - **Positive ex**: Bounded max degree, planar, bounded treewidth, ...
  - **Negative ex**: Cliques, bicliques, ...
- **Attempt 1.** Edge density bounded by a constant:

$$\mathrm{density}(G) := \frac{|E(G)|}{|V(G)|} \leqslant c.$$

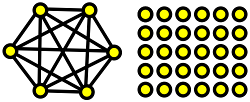
- **Note**: density(G) is half of the average degree.

What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?
  - **Positive ex**: Bounded max degree, planar, bounded treewidth, ...
  - **Negative ex**: Cliques, bicliques, ...
- **Attempt 1.** Edge density bounded by a constant:

$$\text{density}(G) := \frac{|E(G)|}{|V(G)|} \leqslant c.$$

- **Note**: density(G) is half of the average degree.
- **Problem**: Take a clique of size n plus $n^2 - n$ isolated vertices.

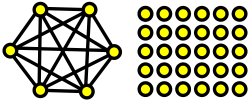


What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?
  - **Positive ex**: Bounded max degree, planar, bounded treewidth, ...
  - **Negative ex**: Cliques, bicliques, ...
- **Attempt 1.** Edge density bounded by a constant:

$$\mathrm{density}(G) := \frac{|E(G)|}{|V(G)|} \leqslant c.$$

- **Note**: density(G) is half of the average degree.
- **Problem**: Take a clique of size n plus $n^2 - n$ isolated vertices.
  - This graph has edge density $< \frac{1}{2}$.

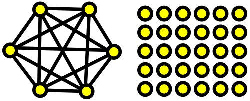


What does it mean sparse?

- **Q**: What does it mean that a graph is sparse?
  - **Positive ex**: Bounded max degree, planar, bounded treewidth, ...
  - **Negative ex**: Cliques, bicliques, ...
- **Attempt 1.** Edge density bounded by a constant:

$$\mathrm{density}(G) := \frac{|E(G)|}{|V(G)|} \leqslant c.$$

- **Note**: density(G) is half of the average degree.
- **Problem**: Take a clique of size n plus $n^2 - n$ isolated vertices.
  - This graph has edge density $< \frac{1}{2}$.
  - **Issue**: Although the density is small, contains a dense substructure.



What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

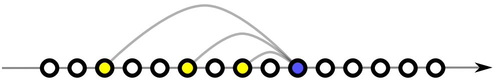
- **Remark**: This is essentially equivalent to **degeneracy**.

What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.

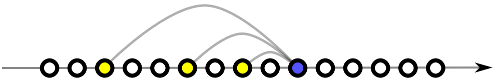


What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.
  - A d-degenerate graph on n vertices has at most dn edges.

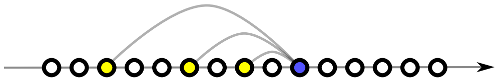


What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.
  - A d-degenerate graph on n vertices has at most dn edges.
    - Hence $\max_{H \subseteq G} \text{density}(H) \leqslant \deg(G)$.

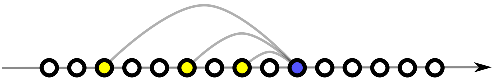


What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.
  - A d-degenerate graph on n vertices has at most dn edges.
    - Hence $\max_{H \subseteq G} \text{density}(H) \leqslant \deg(G)$.
  - If $\text{density}(H) \leqslant c$, then H has a vertex of degree $\leqslant 2c$.

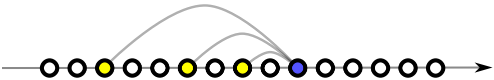


What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

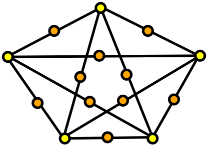
- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.
  - A d-degenerate graph on n vertices has at most dn edges.
    - Hence $\max_{H \subseteq G} \text{density}(H) \leqslant \deg(G)$.
  - If $\text{density}(H) \leqslant c$, then H has a vertex of degree $\leqslant 2c$.
    - Hence $\deg(G) \leqslant 2 \cdot \max_{H \subseteq G} \text{density}(H)$.



- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.
  - A d-degenerate graph on n vertices has at most dn edges.
    - Hence $\max_{H \subseteq G} \text{density}(H) \leqslant \deg(G)$.
  - If $\text{density}(H) \leqslant c$, then H has a vertex of degree $\leqslant 2c$.
    - Hence $\deg(G) \leqslant 2 \cdot \max_{H \subseteq G} \text{density}(H)$.
- **Problem**: Take a clique K_n with each edge subdivided once.

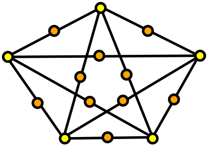


What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.
  - A d-degenerate graph on n vertices has at most dn edges.
    - Hence $\max_{H \subseteq G}$ density$(H) \leqslant \deg(G)$.
  - If density$(H) \leqslant c$, then H has a vertex of degree $\leqslant 2c$.
    - Hence $\deg(G) \leqslant 2 \cdot \max_{H \subseteq G}$ density(H).
- **Problem**: Take a clique K_n with each edge subdivided once.
  - This graph is 2-degenerate $\Rightarrow$ Subgraphs have edge density $\leqslant 2$.

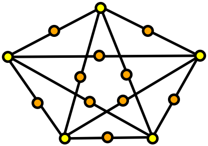


What does it mean sparse?

- **Attempt 2.** Every subgraph of G has bounded edge density:

$$\max_{H \subseteq G} \text{density}(H) \leqslant c.$$

- **Remark**: This is essentially equivalent to **degeneracy**.
  - A graph is d-**degenerate** if it admits a vertex ordering where every vertex has at most d smaller neighbors.
  - A d-degenerate graph on n vertices has at most dn edges.
    - Hence $\max_{H \subseteq G} \text{density}(H) \leqslant \deg(G)$.
  - If $\text{density}(H) \leqslant c$, then H has a vertex of degree $\leqslant 2c$.
    - Hence $\deg(G) \leqslant 2 \cdot \max_{H \subseteq G} \text{density}(H)$.
- **Problem**: Take a clique K_n with each edge subdivided once.
  - This graph is 2-degenerate $\Rightarrow$ Subgraphs have edge density $\leqslant 2$.
  - **Issue**: We see a dense structure "at depth" 1.

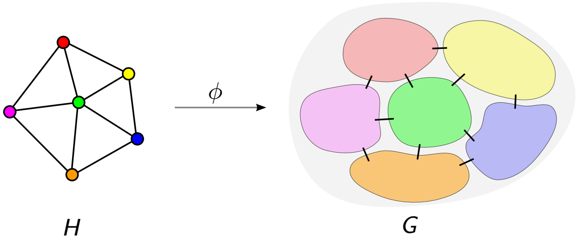


Shallow minors

- **Need**: notion of embedding that looks at constant "depth".

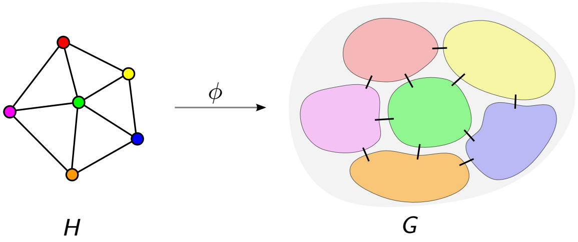
Shallow minors

- **Need**: notion of embedding that looks at constant "depth".
- Graph H is a **minor** of G if there is a **minor model** ϕ of H in G.



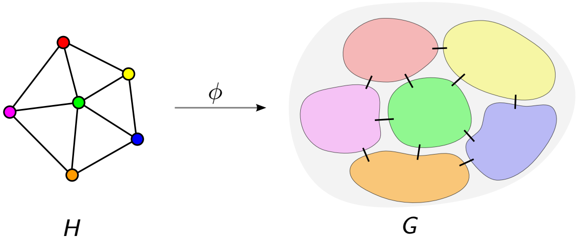
Shallow minors

- **Need**: notion of embedding that looks at constant "depth".
- Graph H is a **minor** of G if there is a **minor model** ϕ of H in G.
  - Model ϕ maps vertices $u \in V(H)$ to pairwise disjoint connected subgraphs $\phi(u)$ of G, called **branch sets**.



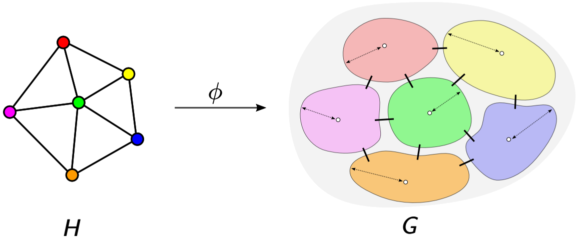
Shallow minors

- **Need**: notion of embedding that looks at constant "depth".
- Graph H is a **minor** of G if there is a **minor model** ϕ of H in G.
  - Model ϕ maps vertices $u \in V(H)$ to pairwise disjoint connected subgraphs $\phi(u)$ of G, called **branch sets**.
  - If $uv \in E(H)$, then there should be an edge between $\phi(u)$ and $\phi(v)$.



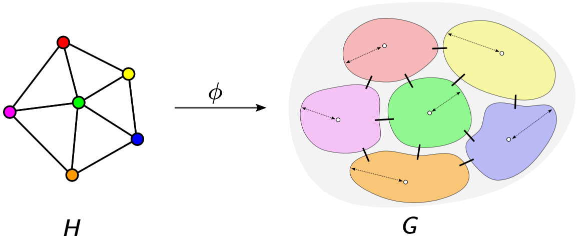
Shallow minors

- **Need**: notion of embedding that looks at constant "depth".
- Graph H is a **minor** of G if there is a **minor model** ϕ of H in G.
  - Model ϕ maps vertices $u \in V(H)$ to pairwise disjoint connected subgraphs $\phi(u)$ of G, called **branch sets**.
  - If $uv \in E(H)$, then there should be an edge between $\phi(u)$ and $\phi(v)$.
- Graph H is a **depth-d minor** of G if there is a minor model of H in G where each branch set has radius at most d.



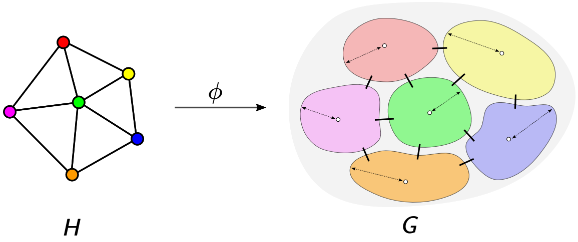
Shallow minors

- **Need**: notion of embedding that looks at constant "depth".
- Graph H is a **minor** of G if there is a **minor model** ϕ of H in G.
  - Model ϕ maps vertices $u \in V(H)$ to pairwise disjoint connected subgraphs $\phi(u)$ of G, called **branch sets**.
  - If $uv \in E(H)$, then there should be an edge between $\phi(u)$ and $\phi(v)$.
- Graph H is a **depth-d minor** of G if there is a minor model of H in G where each branch set has radius at most d.
  - Depth-0 minors $\equiv$ Subgraphs



Shallow minors

- **Need**: notion of embedding that looks at constant "depth".
- Graph H is a **minor** of G if there is a **minor model** ϕ of H in G.
  - Model ϕ maps vertices $u \in V(H)$ to pairwise disjoint connected subgraphs $\phi(u)$ of G, called **branch sets**.
  - If $uv \in E(H)$, then there should be an edge between $\phi(u)$ and $\phi(v)$.
- Graph H is a **depth-d minor** of G if there is a minor model of H in G where each branch set has radius at most d.
  - Depth-0 minors $\equiv$ Subgraphs
- **Idea**: Replace *subgraphs* with *shallow minors* in the def. of sparsity.



Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\nabla d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\nabla d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.
  - **Ex**: $\mathcal{C}\nabla 0$ is the closure of $\mathcal{C}$ under subgraphs.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\nabla d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.
  - **Ex**: $\mathcal{C}\nabla 0$ is the closure of $\mathcal{C}$ under subgraphs.

Bounded expansion

A class of graphs $\mathcal{C}$ has **bounded expansion** if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that density$(H) \leqslant f(d)$ for all $d \in \mathbb{N}$ and $H \in \mathcal{C}\nabla d$.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\nabla d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.
  - **Ex**: $\mathcal{C}\nabla 0$ is the closure of $\mathcal{C}$ under subgraphs.

Bounded expansion

A class of graphs $\mathcal{C}$ has **bounded expansion** if there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that density$(H) \leqslant f(d)$ for all $d \in \mathbb{N}$ and $H \in \mathcal{C}\nabla d$.

Nowhere dense

A class of graphs $\mathcal{C}$ is **nowhere dense** if there is a function $t\colon \mathbb{N} \to \mathbb{N}$ such that $K_{t(d)} \notin \mathcal{C}\nabla d$ for all $d \in \mathbb{N}$.

Equivalently, $\mathcal{C}\nabla d \neq \text{Graphs}$ for all $d \in \mathbb{N}$.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\nabla d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.
  - **Ex**: $\mathcal{C}\nabla 0$ is the closure of $\mathcal{C}$ under subgraphs.

Bounded expansion

A class of graphs $\mathcal{C}$ has **bounded expansion** if there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that density$(H) \leqslant f(d)$ for all $d \in \mathbb{N}$ and $H \in \mathcal{C}\nabla d$.

Nowhere dense

A class of graphs $\mathcal{C}$ is **nowhere dense** if there is a function $t\colon \mathbb{N} \to \mathbb{N}$ such that $K_{t(d)} \notin \mathcal{C}\nabla d$ for all $d \in \mathbb{N}$.

Equivalently, $\mathcal{C}\nabla d \neq \mathrm{Graphs}$ for all $d \in \mathbb{N}$.

- **Intuition**: At every constant depth we see a sparse class, but the parameters can deteriorate with increasing depth.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\nabla d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.
  - **Ex**: $\mathcal{C}\nabla 0$ is the closure of $\mathcal{C}$ under subgraphs.

Bounded expansion

A class of graphs $\mathcal{C}$ has **bounded expansion** if there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that density$(H) \leqslant f(d)$ for all $d \in \mathbb{N}$ and $H \in \mathcal{C}\nabla d$.

Nowhere dense

A class of graphs $\mathcal{C}$ is **nowhere dense** if there is a function $t\colon \mathbb{N} \to \mathbb{N}$ such that $K_{t(d)} \notin \mathcal{C}\nabla d$ for all $d \in \mathbb{N}$.

Equivalently, $\mathcal{C}\nabla d \neq \text{Graphs}$ for all $d \in \mathbb{N}$.

- **Intuition**: At every constant depth we see a sparse class, but the parameters can deteriorate with increasing depth.
- **Note**: Every class of bounded expansion is nowhere dense.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\triangledown d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.
  - **Ex**: $\mathcal{C}\triangledown 0$ is the closure of $\mathcal{C}$ under subgraphs.

Bounded expansion

A class of graphs $\mathcal{C}$ has **bounded expansion** if there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that density$(H) \leqslant f(d)$ for all $d \in \mathbb{N}$ and $H \in \mathcal{C}\triangledown d$.

Nowhere dense

A class of graphs $\mathcal{C}$ is **nowhere dense** if there is a function $t\colon \mathbb{N} \to \mathbb{N}$ such that $K_{t(d)} \notin \mathcal{C}\triangledown d$ for all $d \in \mathbb{N}$.

Equivalently, $\mathcal{C}\triangledown d \neq \mathrm{Graphs}$ for all $d \in \mathbb{N}$.

- **Intuition**: At every constant depth we see a sparse class, but the parameters can deteriorate with increasing depth.
- **Note**: Every class of bounded expansion is nowhere dense.
- **Fact**: Nowhere dense classes are also sparse.

Sparsity, formally

- **Note**: Sparsity is a property of a graph class, not of a single graph.
- **Notation**: $\mathcal{C}\nabla d = \{\text{depth-}d \text{ minors of graphs from } \mathcal{C}\}$.
  - **Ex**: $\mathcal{C}\nabla 0$ is the closure of $\mathcal{C}$ under subgraphs.

Bounded expansion

A class of graphs $\mathcal{C}$ has **bounded expansion** if there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that density$(H) \leqslant f(d)$ for all $d \in \mathbb{N}$ and $H \in \mathcal{C}\nabla d$.

Nowhere dense

A class of graphs $\mathcal{C}$ is **nowhere dense** if there is a function $t\colon \mathbb{N} \to \mathbb{N}$ such that $K_{t(d)} \notin \mathcal{C}\nabla d$ for all $d \in \mathbb{N}$.

Equivalently, $\mathcal{C}\nabla d \neq \mathrm{Graphs}$ for all $d \in \mathbb{N}$.

- **Intuition**: At every constant depth we see a sparse class, but the parameters can deteriorate with increasing depth.
- **Note**: Every class of bounded expansion is nowhere dense.
- **Fact**: Nowhere dense classes are also sparse.
  - If $H \in \mathcal{C}\nabla d$, then H has $\mathcal{O}_{\varepsilon,d}(n^{1+\varepsilon})$ edges, for any $\varepsilon > 0$.

Hierarchy of sparsity

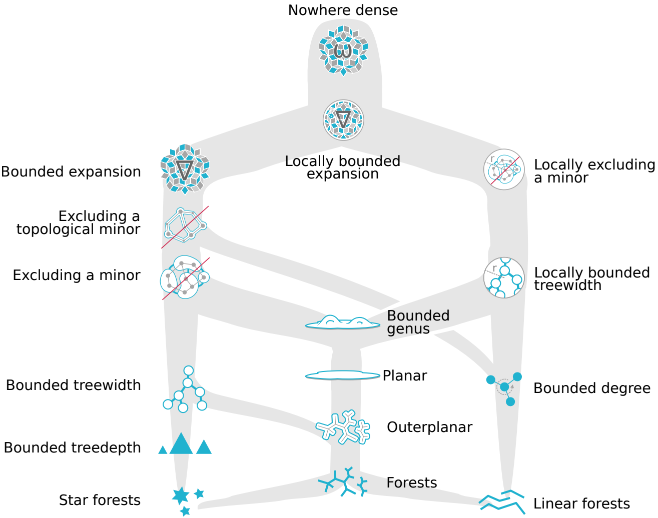


Figure by Felix Reidl

Theory of sparsity

- Developed by Nešetřil and Ossona de Mendez since 2005.

- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.

- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.
- Many concepts appeared already much earlier.

- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.
- Many concepts appeared already much earlier.
  - Earliest definition of nowhere denseness: Podewski and Ziegler in 1976.

Theory of sparsity

- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.
- Many concepts appeared already much earlier.
  - Earliest definition of nowhere denseness: Podewski and Ziegler in 1976.
- **In summary**:

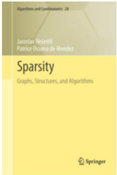
Theory of sparsity

- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.

- Many concepts appeared already much earlier.
  - Earliest definition of nowhere denseness:
    Podewski and Ziegler in 1976.
- **In summary**:
  - Bounded expansion and nowhere denseness are fundamental concepts that have multiple equivalent characterizations.

Theory of sparsity

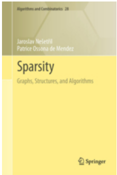
- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.



- Many concepts appeared already much earlier.
  - Earliest definition of nowhere denseness:
    Podewski and Ziegler in 1976.
- **In summary**:
  - Bounded expansion and nowhere denseness are fundamental concepts that have multiple equivalent characterizations.
  - Each characterization yields a different viewpoint and a tool.

Theory of sparsity

- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.
- Many concepts appeared already much earlier.
  - Earliest definition of nowhere denseness:
    Podewski and Ziegler in 1976.



- **In summary**:
  - Bounded expansion and nowhere denseness are fundamental concepts that have multiple equivalent characterizations.
  - Each characterization yields a different viewpoint and a tool.
  - Applications for **combinatorics**, **algorithms**, and **logic**.

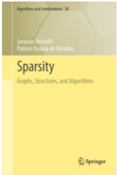
Theory of sparsity

- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.
- Many concepts appeared already much earlier.
  - Earliest definition of nowhere denseness: Podewski and Ziegler in 1976.

- **In summary**:
  - Bounded expansion and nowhere denseness are fundamental concepts that have multiple equivalent characterizations.
  - Each characterization yields a different viewpoint and a tool.
  - Applications for **combinatorics**, **algorithms**, and **logic**.
  - Nowhere denseness delimits tractability for many basic problems.

Theory of sparsity

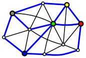
- Developed by Nešetřil and Ossona de Mendez since 2005.
  - Monograph *Sparsity* presents the field as of 2012.
- Many concepts appeared already much earlier.
  - Earliest definition of nowhere denseness:
    Podewski and Ziegler in 1976.



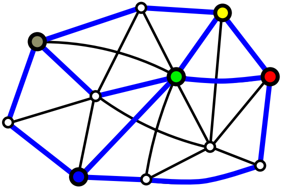
- **In summary**:
  - Bounded expansion and nowhere denseness are fundamental concepts that have multiple equivalent characterizations.
  - Each characterization yields a different viewpoint and a tool.
  - Applications for **combinatorics**, **algorithms**, and **logic**.
  - Nowhere denseness delimits tractability for many basic problems.
  - Toolbox seems much more suitable than using decomposition theorems for classes excluding a fixed (topological) minor.

Sparsity of shallow minors

Sparsity of shallow topological minors

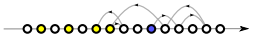


Sparsity of shallow minors

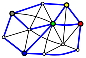


Characterizations

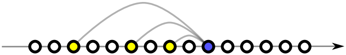
Generalized coloring numbers



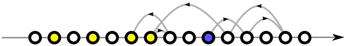
Sparsity of shallow topological minors



Sparsity of shallow minors



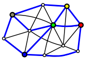
Degeneracy



Weak coloring number

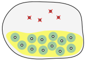
Generalized coloring numbers

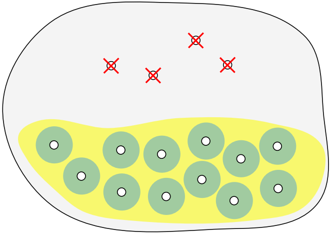
Sparsity of shallow topological minors



Sparsity of shallow minors

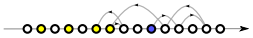
Uniform quasi-wideness



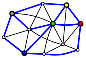


Characterizations

Generalized coloring numbers

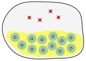


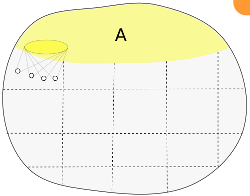
Sparsity of shallow topological minors



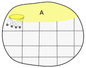
Sparsity of shallow minors

Uniform quasi-wideness



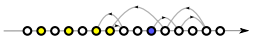


Neighborhood complexity

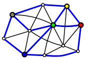


Characterizations

Generalized coloring numbers



Sparsity of shallow topological minors



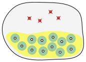
Fraternal augmentations

Sparsity of shallow minors

Neighborhood covers

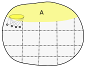
Low treedepth colorings

Uniform quasi-wideness



k-Helly property

Neighborhood complexity



Splitter game

Applications

- The toolbox of sparsity can be applied in various paradigms:

Applications

- The toolbox of sparsity can be applied in various paradigms:
  - **parameterized algorithms**, including **kernelization**;

Applications

- The toolbox of sparsity can be applied in various paradigms:
  - **parameterized algorithms**, including **kernelization**;
  - approximation algorithms;

Applications

- The toolbox of sparsity can be applied in various paradigms:
  - **parameterized algorithms**, including **kernelization**;
  - approximation algorithms;
  - distributed algorithms;

Applications

- The toolbox of sparsity can be applied in various paradigms:
  - **parameterized algorithms**, including **kernelization**;
  - approximation algorithms;
  - distributed algorithms;
  - model-checking and logic in general;

Applications

- The toolbox of sparsity can be applied in various paradigms:
  - **parameterized algorithms**, including **kernelization**;
  - approximation algorithms;
  - distributed algorithms;
  - model-checking and logic in general;
  - circuit complexity;

Applications

- The toolbox of sparsity can be applied in various paradigms:
  - **parameterized algorithms**, including **kernelization**;
  - approximation algorithms;
  - distributed algorithms;
  - model-checking and logic in general;
  - circuit complexity;
  - dynamic problems...

Applications

- The toolbox of sparsity can be applied in various paradigms:
  - **parameterized algorithms**, including **kernelization**;
  - approximation algorithms;
  - distributed algorithms;
  - model-checking and logic in general;
  - circuit complexity;
  - dynamic problems...
- Inherently applies to problems with **local character**.

Local problems: algorithmics

r-Scattered Set

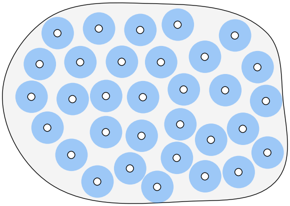
I Graph G and integer k

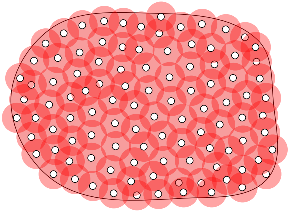
Q Is there $I \subseteq V(G)$ with $|I| = k$ s.t. r-balls around vertices of I are disjoint, equivalently vertices of I are pairwise at distance $> 2r$?

r-Dominating Set

I Graph G and integer k

Q Is there $D \subseteq V(G)$ with $|D| = k$ s.t. every vertex of G is at distance $\leqslant r$ from some vertex of D?





Local problems: logic

- First order logic (FO) on graphs:

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that u_i and u_j are non-adjacent for all $i \neq j$.

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that u_i and u_j are non-adjacent for all $i \neq j$.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that for every v, the vertex v is either equal or adjacent to one of $u_1, \ldots, u_k$.

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that u_i and u_j are non-adjacent for all $i \neq j$.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that for every v, the vertex v is either equal or adjacent to one of $u_1, \ldots, u_k$.

FO model-checking

Input	Graph G, FO sentence φ
Question	Is φ satisfied in G?

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that u_i and u_j are non-adjacent for all $i \neq j$.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that for every v, the vertex v is either equal or adjacent to one of $u_1, \ldots, u_k$.

FO model-checking

Input	Graph G, FO sentence φ
Question	Is φ satisfied in G?

- In general graphs there is an $\mathcal{O}(n^{\|\varphi\|})$-time algorithm, but the problem is AW[$\star$]-complete.

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that u_i and u_j are non-adjacent for all $i \neq j$.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that for every v, the vertex v is either equal or adjacent to one of $u_1, \ldots, u_k$.

FO model-checking

Input	Graph G, FO sentence φ
Question	Is φ satisfied in G?

- In general graphs there is an $\mathcal{O}(n^{\|\varphi\|})$-time algorithm, but the problem is AW[$\star$]-complete.
- FPT algorithms for bounded degree, planar, H-minor-free, ...

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that u_i and u_j are non-adjacent for all $i \neq j$.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that for every v, the vertex v is either equal or adjacent to one of $u_1, \ldots, u_k$.

FO model-checking

Input	Graph G, FO sentence φ
Question	Is φ satisfied in G?

- In general graphs there is an $\mathcal{O}(n^{\|\varphi\|})$-time algorithm, but the problem is AW[$\star$]-complete.
- FPT algorithms for bounded degree, planar, H-minor-free, ...
  - **Crux**: FO is **local** — a formula of quantifier rank q cannot measure distances larger than 2^q.

Local problems: logic

- First order logic (FO) on graphs:
  - We may quantify vrts existentially or universally and test adjacency.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that u_i and u_j are non-adjacent for all $i \neq j$.
  - **Ex**: There exist $u_1, \ldots, u_k$ such that for every v, the vertex v is either equal or adjacent to one of $u_1, \ldots, u_k$.

FO model-checking

Input	Graph G, FO sentence φ
Question	Is φ satisfied in G?

- In general graphs there is an $\mathcal{O}(n^{\|\varphi\|})$-time algorithm, but the problem is AW[$\star$]-complete.
- FPT algorithms for bounded degree, planar, H-minor-free, ...
  - **Crux**: FO is **local** — a formula of quantifier rank q cannot measure distances larger than 2^q.
- Same story for r-Dominating Set and r-Independent Set.

Dichotomies

Theorem [Grohe et al., Dvořák et al.]

Let $\mathcal{C}$ be a monotone graph class (closed under taking subgraphs). Then:

- If $\mathcal{C}$ is nowhere dense, then FO model-checking can be done in time $f(\|\varphi\|) \cdot n^{1+\varepsilon}$ on graphs from $\mathcal{C}$, for any $\varepsilon > 0$.
- If $\mathcal{C}$ is somewhere dense, then FO model-checking is AW[$\star$]-complete on graphs from $\mathcal{C}$.

Dichotomies

Theorem [Grohe et al., Dvořák et al.]

Let $\mathcal{C}$ be a monotone graph class (closed under taking subgraphs). Then:

- If $\mathcal{C}$ is nowhere dense, then FO model-checking can be done in time $f(\|\varphi\|) \cdot n^{1+\varepsilon}$ on graphs from $\mathcal{C}$, for any $\varepsilon > 0$.
- If $\mathcal{C}$ is somewhere dense, then FO model-checking is AW[$\star$]-complete on graphs from $\mathcal{C}$.

Theorem [Drange et al., Eickmeyer et al.]

Let $\mathcal{C}$ be a monotone graph class. Then:

- If $\mathcal{C}$ is nowhere dense, then for every $r \in \mathbb{N}$ and $\varepsilon > 0$, r-DomSet admits a kernel with $\mathcal{O}(k^{1+\varepsilon})$ vertices on graphs from $\mathcal{C}$.
- If $\mathcal{C}$ is somewhere dense, then for some $r \in \mathbb{N}$, r-DomSet is W[2]-hard on graphs from $\mathcal{C}$.

Dichotomies

Theorem [Grohe et al., Dvořák et al.]

Let $\mathcal{C}$ be a monotone graph class (closed under taking subgraphs). Then:

- If $\mathcal{C}$ is nowhere dense, then FO model-checking can be done in time $f(\|\varphi\|) \cdot n^{1+\varepsilon}$ on graphs from $\mathcal{C}$, for any $\varepsilon > 0$.
- If $\mathcal{C}$ is somewhere dense, then FO model-checking is AW[$\star$]-complete on graphs from $\mathcal{C}$.

Theorem [Drange et al., Eickmeyer et al.]

Let $\mathcal{C}$ be a monotone graph class. Then:

- If $\mathcal{C}$ is nowhere dense, then for every $r \in \mathbb{N}$ and $\varepsilon > 0$, r-$\textsc{DomSet}$ admits a kernel with $\mathcal{O}(k^{1+\varepsilon})$ vertices on graphs from $\mathcal{C}$.
- If $\mathcal{C}$ is somewhere dense, then for some $r \in \mathbb{N}$, r-$\textsc{DomSet}$ is W[2]-hard on graphs from $\mathcal{C}$.

- Nowhere denseness **exactly** characterizes monotone classes where local problems are tractable from the parameterized viewpoint.

Uniform quasi-wideness

- **Intuition**: In a large sparse graph there should be many vertices pairwise far from each other.

Uniform quasi-wideness

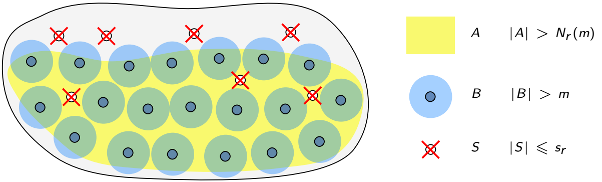
- **Intuition**: In a large sparse graph there should be many vertices pairwise far from each other.
- **WRONG**: a star.

Uniform quasi-wideness

- **Intuition**: In a large sparse graph there should be many vertices pairwise far from each other.
- **WRONG**: a star.

Uniform quasi-wideness

Class $\mathcal{C}$ is **uniformly quasi-wide** if

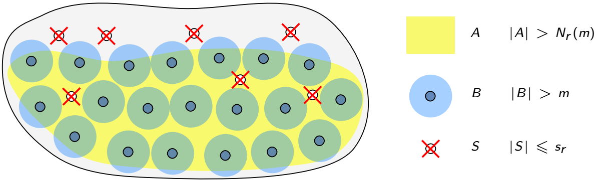


Uniform quasi-wideness

- **Intuition**: In a large sparse graph there should be many vertices pairwise far from each other.
- **WRONG**: a star.

Uniform quasi-wideness

Class $\mathcal{C}$ is **uniformly quasi-wide** if for every $r \in \mathbb{N}$

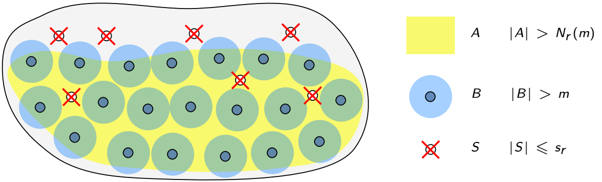


Uniform quasi-wideness

- **Intuition**: In a large sparse graph there should be many vertices pairwise far from each other.
- **WRONG**: a star.

Uniform quasi-wideness

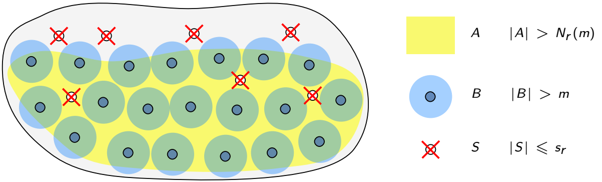
Class $\mathcal{C}$ is **uniformly quasi-wide** if for every $r \in \mathbb{N}$ there exist $s_r \in \mathbb{N}$ and $N_r : \mathbb{N} \to \mathbb{N}$ s.t.



- **Intuition**: In a large sparse graph there should be many vertices pairwise far from each other.
- **WRONG**: a star.

Uniform quasi-wideness

Class $\mathcal{C}$ is **uniformly quasi-wide** if for every $r \in \mathbb{N}$ there exist $s_r \in \mathbb{N}$ and $N_r : \mathbb{N} \to \mathbb{N}$ s.t. for every graph $G \in \mathcal{C}$, $m \in \mathbb{N}$, and vertex subset $A \subseteq V(G)$ of size larger than $N_r(m)$,



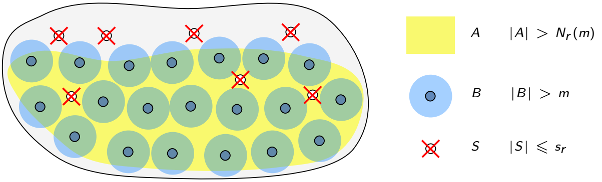
Uniform quasi-wideness

- **Intuition**: In a large sparse graph there should be many vertices pairwise far from each other.
- **WRONG**: a star.

Uniform quasi-wideness

Class $\mathcal{C}$ is **uniformly quasi-wide** if for every $r \in \mathbb{N}$ there exist $s_r \in \mathbb{N}$ and $N_r : \mathbb{N} \to \mathbb{N}$ s.t. for every graph $G \in \mathcal{C}$, $m \in \mathbb{N}$, and vertex subset $A \subseteq V(G)$ of size larger than $N_r(m)$, there exist sets $S \subseteq V(G)$ and $B \subseteq A - S$ with the following properties:

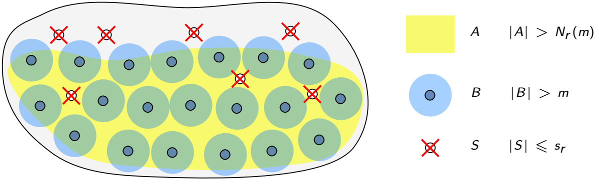
- $|S| \leqslant s_r$, $|B| > m$, and
- B is r-scattered in $G - S$.



Uniform quasi-wideness

Theorem [Nešetřil and Ossona de Mendez]

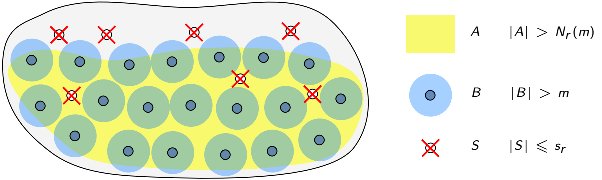
A class of graphs is uniformly quasi-wide iff it is nowhere dense.



Theorem [Nešetřil and Ossona de Mendez]

A class of graphs is uniformly quasi-wide iff it is nowhere dense.

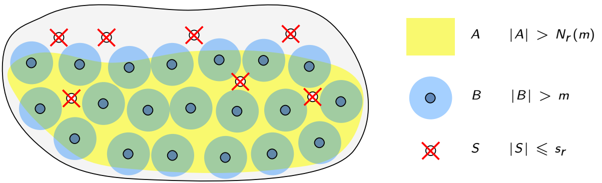
- **Remark**: For fixed $\mathcal{C}$ and r, $N_r(m)$ is a polynomial in m.



Theorem [Nešetřil and Ossona de Mendez]

A class of graphs is uniformly quasi-wide iff it is nowhere dense.

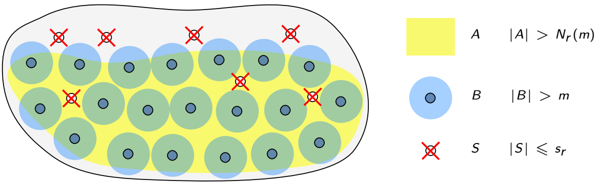
- **Remark**: For fixed $\mathcal{C}$ and r, $N_r(m)$ is a polynomial in m.
- Given (G, A), sets S and B can be found in time $\mathrm{poly}(m) \cdot |G|$.



Theorem [Nešetřil and Ossona de Mendez]

A class of graphs is uniformly quasi-wide iff it is nowhere dense.

- **Remark**: For fixed $\mathcal{C}$ and r, $N_r(m)$ is a polynomial in m.
- Given (G, A), sets S and B can be found in time $\mathrm{poly}(m) \cdot |G|$.
- Useful combinatorial tool for "digging structure" in sparse graphs.

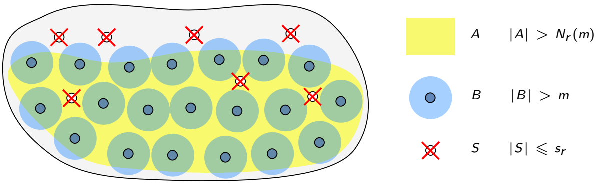


Uniform quasi-wideness

Theorem [Nešetřil and Ossona de Mendez]

A class of graphs is uniformly quasi-wide iff it is nowhere dense.

- **Remark**: For fixed $\mathcal{C}$ and r, $N_r(m)$ is a polynomial in m.
- Given (G, A), sets S and B can be found in time $\mathrm{poly}(m) \cdot |G|$.
- Useful combinatorial tool for "digging structure" in sparse graphs.
- **Now**: a simple FPT algorithm for r-Dominating Set in any nowhere dense class $\mathcal{C}$.

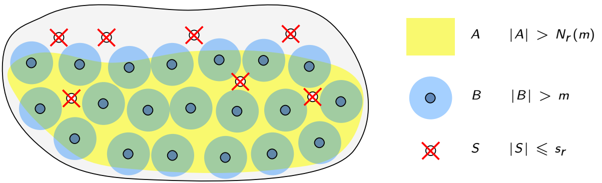


Uniform quasi-wideness

Theorem [Nešetřil and Ossona de Mendez]

A class of graphs is uniformly quasi-wide iff it is nowhere dense.

- **Remark**: For fixed $\mathcal{C}$ and r, $N_r(m)$ is a polynomial in m.
- Given (G, A), sets S and B can be found in time $\mathrm{poly}(m) \cdot |G|$.
- Useful combinatorial tool for "digging structure" in sparse graphs.
- **Now**: a simple FPT algorithm for r-Dominating Set in any nowhere dense class $\mathcal{C}$.
  - Based on an unpublished work with S. Siebertz and Sz. Toruńczyk.



Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.
  - **Runtime**: $2^{\mathcal{O}(k|A|)} \cdot (n+m)$, as the number of classes is $2^{|A|}$ at worst.

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.
  - **Runtime**: $2^{\mathcal{O}(k|A|)} \cdot (n+m)$, as the number of classes is $2^{|A|}$ at worst.
  - **Remark**: If G is from a nowhere dense class $\mathcal{C}$, then the number of classes is actually $\mathcal{O}(|A|^{1+\varepsilon})$ for any $\varepsilon > 0$.

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.
  - **Runtime**: $2^{\mathcal{O}(k|A|)} \cdot (n+m)$, as the number of classes is $2^{|A|}$ at worst.
  - **Remark**: If G is from a nowhere dense class $\mathcal{C}$, then the number of classes is actually $\mathcal{O}(|A|^{1+\varepsilon})$ for any $\varepsilon > 0$.
  - **Runtime**: $|A|^{\mathcal{O}(k)} \cdot (n+m)$.

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.
  - **Runtime**: $2^{\mathcal{O}(k|A|)} \cdot (n+m)$, as the number of classes is $2^{|A|}$ at worst.
  - **Remark**: If G is from a nowhere dense class $\mathcal{C}$, then the number of classes is actually $\mathcal{O}(|A|^{1+\varepsilon})$ for any $\varepsilon > 0$.
  - **Runtime**: $|A|^{\mathcal{O}(k)} \cdot (n+m)$.
- **Dynamic programming** on subsets of A:

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.
  - **Runtime**: $2^{\mathcal{O}(k|A|)} \cdot (n+m)$, as the number of classes is $2^{|A|}$ at worst.
  - **Remark**: If G is from a nowhere dense class $\mathcal{C}$, then the number of classes is actually $\mathcal{O}(|A|^{1+\varepsilon})$ for any $\varepsilon > 0$.
  - **Runtime**: $|A|^{\mathcal{O}(k)} \cdot (n+m)$.
- **Dynamic programming** on subsets of A:
  - For $X \subseteq A$, let $\mathrm{D}[X]$ be the smallest number of vertices that together r-dominate X.

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.
  - **Runtime**: $2^{\mathcal{O}(k|A|)} \cdot (n+m)$, as the number of classes is $2^{|A|}$ at worst.
  - **Remark**: If G is from a nowhere dense class $\mathcal{C}$, then the number of classes is actually $\mathcal{O}(|A|^{1+\varepsilon})$ for any $\varepsilon > 0$.
  - **Runtime**: $|A|^{\mathcal{O}(k)} \cdot (n+m)$.
- **Dynamic programming** on subsets of A:
  - For $X \subseteq A$, let $\mathrm{D}[X]$ be the smallest number of vertices that together r-dominate X.
  - Then $\mathrm{D}[\emptyset] = 0$ and

$$\mathrm{D}[X] = 1 + \min_{v \in V(G)} \mathrm{D}[X - (N_r[v] \cap A)]$$

Dominating a bounded number of vertices

- Suppose we have a set $A \subseteq V(G)$ whose size is small. We want to check whether one can r-dominate A using at most k vertices.
- **Naive approach**:
  - Classify vertices $u \in V(G)$ according to $N_r[u] \cap A$, i.e., what they r-dominate in A.
  - Vertices in the same class are interchangable.
  - Examine all k-tuples of classes checking if any covers whole A.
  - **Runtime**: $2^{\mathcal{O}(k|A|)} \cdot (n+m)$, as the number of classes is $2^{|A|}$ at worst.
  - **Remark**: If G is from a nowhere dense class $\mathcal{C}$, then the number of classes is actually $\mathcal{O}(|A|^{1+\varepsilon})$ for any $\varepsilon > 0$.
  - **Runtime**: $|A|^{\mathcal{O}(k)} \cdot (n+m)$.
- **Dynamic programming** on subsets of A:
  - For $X \subseteq A$, let $\mathrm{D}[X]$ be the smallest number of vertices that together r-dominate X.
  - Then $\mathrm{D}[\emptyset] = 0$ and

$$\mathrm{D}[X] = 1 + \min_{v \in V(G)} \mathrm{D}[X - (N_r[v] \cap A)]$$

  - **Runtime**: $2^{|A|} \cdot |A|^{\mathcal{O}(1)} \cdot (n+m)$.

Simple algorithm for r-DomSet

Simple algorithm for r-DomSet

D_1 

Round 1: Take any k-tuple of vertices D_1.

Simple algorithm for r-DomSet

D_1 

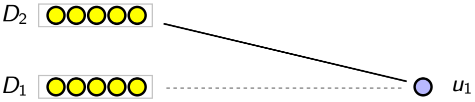
Round 1: Take any k-tuple of vertices D_1.

If D_1 is an r-domset, terminate.

Round 1: Take any k-tuple of vertices D_1.

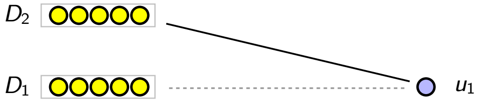
If D_1 is an r-domset, terminate.

Otherwise there is an undominated vertex u_1.



Round 2: Find any k-tuple of vertices D_2 that r-dominates u_1.

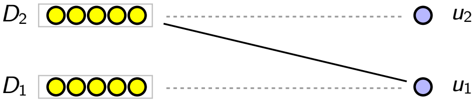
Simple algorithm for r-DomSet



Round 2: Find any k-tuple of vertices D_2 that r-dominates u_1.

If D_2 is an r-domset, terminate.

Simple algorithm for r-DomSet

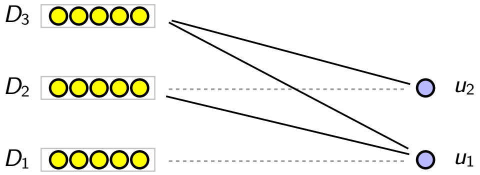


Round 2: Find any k-tuple of vertices D_2 that r-dominates u_1.

If D_2 is an r-domset, terminate.

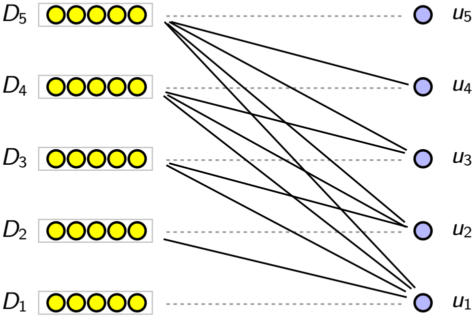
Otherwise there is an undominated vertex u_2.

Simple algorithm for r-DomSet



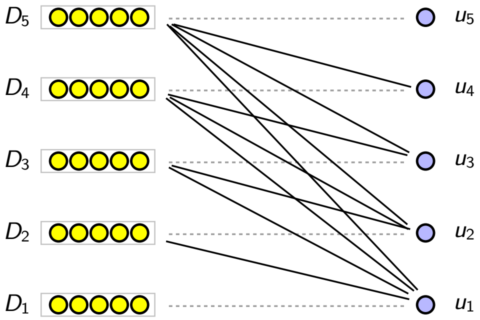
Round 3: Find any k-tuple of vertices D_3 that r-dominates $\{u_1, u_2\}$.

Simple algorithm for r-DomSet



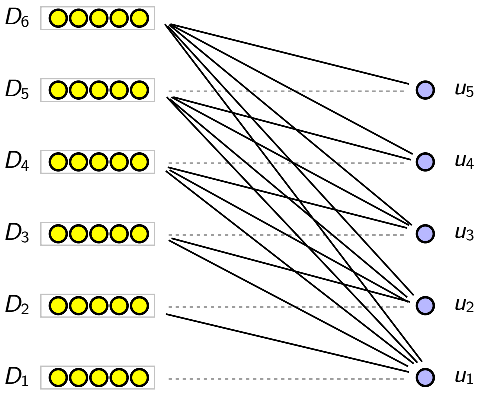
Round i: Find any k-tuple of vertices D_i that r-dominates $\{u_1, \ldots, u_{i-1}\}$.

Simple algorithm for r-DomSet



Round i: Find any k-tuple of vertices D_i that r-dominates $\{u_1, \ldots, u_{i-1}\}$.

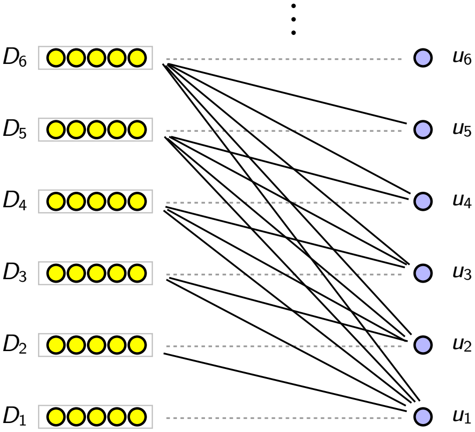
If there is none, answer **NO**.



Round i: Find any k-tuple of vertices D_i that r-dominates $\{u_1, \ldots, u_{i-1}\}$.

If there is none, answer **NO**. If D_i is an r-domset, answer **YES**.

Simple algorithm for r-DomSet



Round i: Find any k-tuple of vertices D_i that r-dominates $\{u_1, \ldots, u_{i-1}\}$.

If there is none, answer **NO**.

If D_i is an r-domset, answer **YES**.

Otherwise there is an undominated vertex u_i. **Proceed**

Analysis

- Fix $r \in \mathbb{N}$ and a nowhere dense graph class $\mathcal{C}$ from which the input graph G is drawn.

Analysis

- Fix $r \in \mathbb{N}$ and a nowhere dense graph class $\mathcal{C}$ from which the input graph G is drawn.
- Each round may either terminate the algorithm or proceed.

Analysis

- Fix $r \in \mathbb{N}$ and a nowhere dense graph class $\mathcal{C}$ from which the input graph G is drawn.
- Each round may either terminate the algorithm or proceed.
- **Naive**: Round i can be implemented in time $i^{\mathcal{O}(k)} \cdot (n+m)$.

Analysis

- Fix $r \in \mathbb{N}$ and a nowhere dense graph class $\mathcal{C}$ from which the input graph G is drawn.
- Each round may either terminate the algorithm or proceed.
- **Naive**: Round i can be implemented in time $i^{\mathcal{O}(k)} \cdot (n+m)$.
- A priori there is no upper bound on the number of rounds.

Analysis

- Fix $r \in \mathbb{N}$ and a nowhere dense graph class $\mathcal{C}$ from which the input graph G is drawn.
- Each round may either terminate the algorithm or proceed.
- **Naive**: Round i can be implemented in time $i^{\mathcal{O}(k)} \cdot (n+m)$.
- A priori there is no upper bound on the number of rounds.

Theorem

There exists a constant c, depending only on $\mathcal{C}$ and $r \in \mathbb{N}$, such that the Algorithm terminates after at most $c \cdot k^c$ rounds.

Analysis

- Fix $r \in \mathbb{N}$ and a nowhere dense graph class $\mathcal{C}$ from which the input graph G is drawn.
- Each round may either terminate the algorithm or proceed.
- **Naive**: Round i can be implemented in time $i^{\mathcal{O}(k)} \cdot (n+m)$.
- A priori there is no upper bound on the number of rounds.

Theorem

There exists a constant c, depending only on $\mathcal{C}$ and $r \in \mathbb{N}$, such that the Algorithm terminates after at most $c \cdot k^c$ rounds.

Corollary

For every fixed nowhere dense class $\mathcal{C}$ and $r \in \mathbb{N}$, the r-$\textsc{DomSet}$ problem on graphs from $\mathcal{C}$ can be solved in time $k^{\mathcal{O}(k)} \cdot (n+m)$.

Analysis

- Fix $r \in \mathbb{N}$ and a nowhere dense graph class $\mathcal{C}$ from which the input graph G is drawn.
- Each round may either terminate the algorithm or proceed.
- **Naive**: Round i can be implemented in time $i^{\mathcal{O}(k)} \cdot (n+m)$.
- A priori there is no upper bound on the number of rounds.

Theorem

There exists a constant c, depending only on $\mathcal{C}$ and $r \in \mathbb{N}$, such that the Algorithm terminates after at most $c \cdot k^c$ rounds.

Corollary

For every fixed nowhere dense class $\mathcal{C}$ and $r \in \mathbb{N}$, the r-DomSet problem on graphs from $\mathcal{C}$ can be solved in time $k^{\mathcal{O}(k)} \cdot (n+m)$.

- **Note**: The algorithm is oblivious of the class $\mathcal{C}$, it may be applied on any input graph G and parameter k.

Case $k = 1$

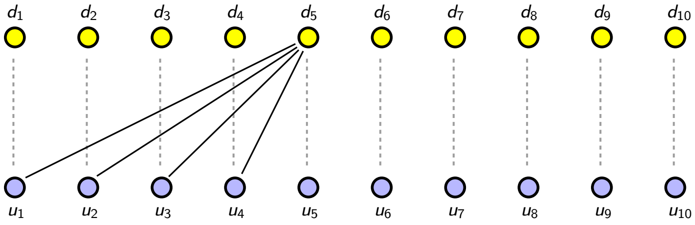
- Let $s_r \in \mathbb{N}$ and $N_r : \mathbb{N} \to \mathbb{N}$ witness uniform quasi-wideness of $\mathcal{C}$.

Case $k = 1$

- Let $s_r \in \mathbb{N}$ and $N_r : \mathbb{N} \to \mathbb{N}$ witness uniform quasi-wideness of $\mathcal{C}$.
- Suppose the algorithm performed more than

$$\ell = N_r(2(r+2)^{s_r})$$

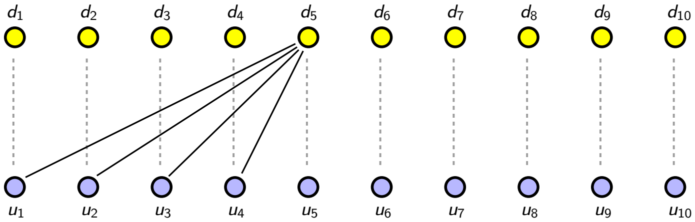
rounds, constructing a **quasi-ladder** of length more than ℓ.



- Let $s_r \in \mathbb{N}$ and $N_r \colon \mathbb{N} \to \mathbb{N}$ witness uniform quasi-wideness of $\mathcal{C}$.
- Suppose the algorithm performed more than

$$\ell = N_r(2(r+2)^{s_r})$$

rounds, constructing a **quasi-ladder** of length more than ℓ.
- Let $A = \{u_1, \ldots, u_{\ell+1}\}$. Apply uniform quasi-wideness to A:

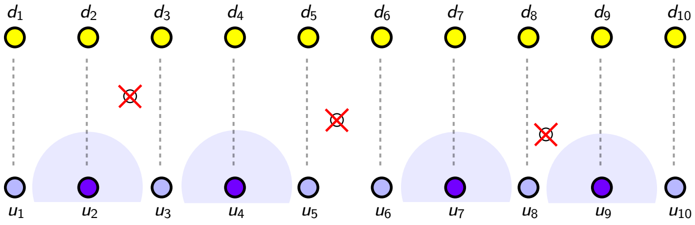


Case $k = 1$

- Let $s_r \in \mathbb{N}$ and $N_r : \mathbb{N} \to \mathbb{N}$ witness uniform quasi-wideness of $\mathcal{C}$.
- Suppose the algorithm performed more than

$$\ell = N_r(2(r+2)^{s_r})$$

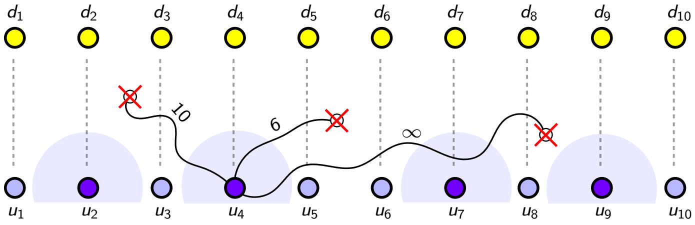
  rounds, constructing a **quasi-ladder** of length more than ℓ.
- Let $A = \{u_1, \ldots, u_{\ell+1}\}$. Apply uniform quasi-wideness to A:
  - Disjoint sets S and $B \subseteq A$ with $|S| \leqslant s_r$ and $|B| > 2(r+2)^{s_r}$, such that B is r-scattered in $G - S$.



Case $k = 1$

- For $u_i \in B$, let $f_i \colon S \to \{0, \ldots, r, \infty\}$ be its r-**distance profile** on S:

$$f_i(v) = \begin{cases} \mathrm{dist}(u_i, v) & \text{if it is } \leqslant r; \\ \infty & \text{otherwise.} \end{cases}$$

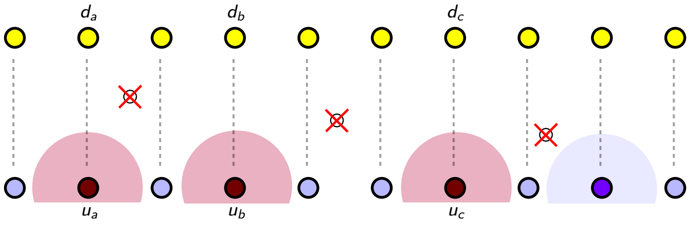


Case $k = 1$

- For $u_i \in B$, let $f_i \colon S \to \{0, \ldots, r, \infty\}$ be its r-**distance profile** on S:

$$f_i(v) = \begin{cases} \operatorname{dist}(u_i, v) & \text{if it is } \leqslant r; \\ \infty & \text{otherwise.} \end{cases}$$

- There are $(r+2)^{s_r}$ possible profiles, so since $|B| > 2(r+2)^{s_r}$, we have $a < b < c$ such that u_a, u_b, u_c have the same profile.

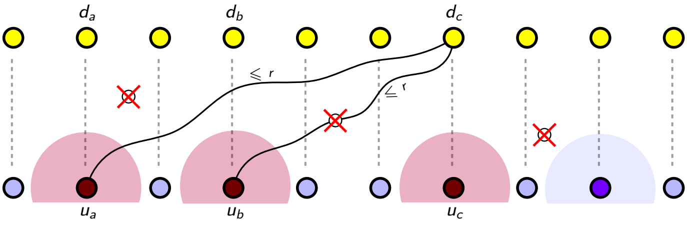


Case $k = 1$

- For $u_i \in B$, let $f_i \colon S \to \{0, \ldots, r, \infty\}$ be its r-**distance profile** on S:

$$f_i(v) = \begin{cases} \mathrm{dist}(u_i, v) & \text{if it is } \leqslant r; \\ \infty & \text{otherwise.} \end{cases}$$

- There are $(r+2)^{s_r}$ possible profiles, so since $|B| > 2(r+2)^{s_r}$, we have $a < b < c$ such that u_a, u_b, u_c have the same profile.
- **Recall**: d_c r-dominates u_a and u_b.

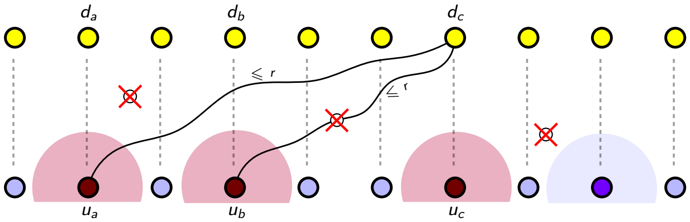


Case $k = 1$

- For $u_i \in B$, let $f_i \colon S \to \{0, \ldots, r, \infty\}$ be its r-**distance profile** on S:

$$f_i(v) = \begin{cases} \mathrm{dist}(u_i, v) & \text{if it is } \leqslant r; \\ \infty & \text{otherwise.} \end{cases}$$

- There are $(r+2)^{s_r}$ possible profiles, so since $|B| > 2(r+2)^{s_r}$, we have $a < b < c$ such that u_a, u_b, u_c have the same profile.
- **Recall**: d_c r-dominates u_a and u_b.
- Since B is r-scattered in $G - S$, at least one of these r-domination paths must cross S; say the one to u_b.

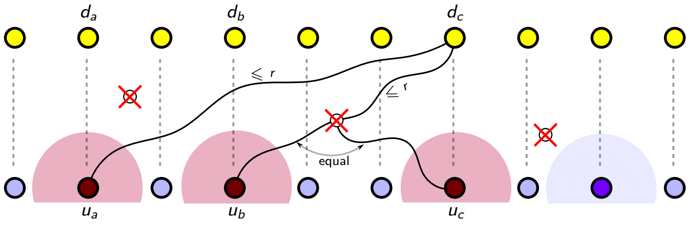


Case $k = 1$

- For $u_i \in B$, let $f_i \colon S \to \{0, \ldots, r, \infty\}$ be its r-**distance profile** on S:

$$f_i(v) = \begin{cases} \mathrm{dist}(u_i, v) & \text{if it is } \leqslant r; \\ \infty & \text{otherwise.} \end{cases}$$

- There are $(r+2)^{s_r}$ possible profiles, so since $|B| > 2(r+2)^{s_r}$, we have $a < b < c$ such that u_a, u_b, u_c have the same profile.
- **Recall**: d_c r-dominates u_a and u_b.
- Since B is r-scattered in $G - S$, at least one of these r-domination paths must cross S; say the one to u_b.
- u_b and u_c have the same profile $\Rightarrow \mathrm{dist}(d_c, u_c) \leqslant r$, contradiction.



Case $k > 1$

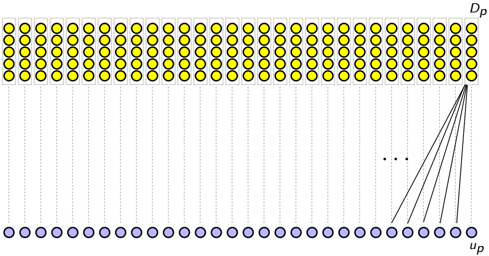
- For $k = 1$ the number of rounds is at most ℓ.

Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.

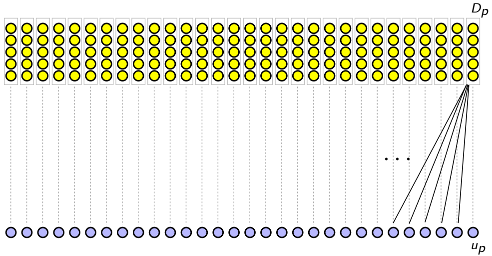
Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.
- Suppose the Algorithm performed $p = k^{\ell+1}$ rounds.



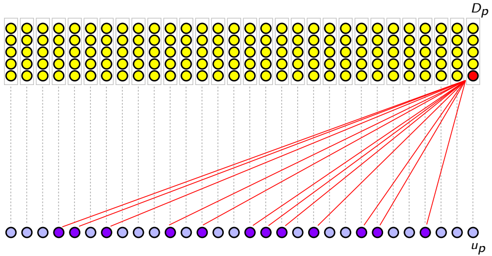
Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.
- Suppose the Algorithm performed $p = k^{\ell+1}$ rounds.
- Examine D_p:



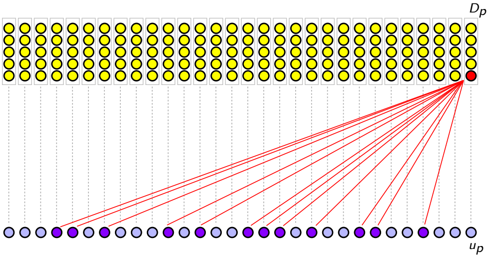
Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.
- Suppose the Algorithm performed $p = k^{\ell+1}$ rounds.
- Examine D_p:
  - Some $d_p \in D_p$ r-dominates $\geqslant \frac{1}{k}$ fraction of $\{u_1, \ldots, u_{p-1}\}$.



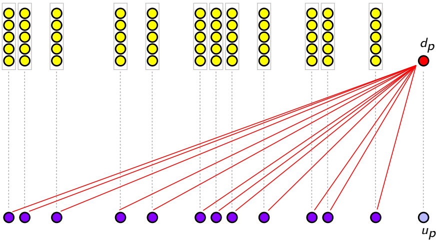
Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.
- Suppose the Algorithm performed $p = k^{\ell+1}$ rounds.
- Examine D_p:
  - Some $d_p \in D_p$ r-dominates $\geqslant \frac{1}{k}$ fraction of $\{u_1, \ldots, u_{p-1}\}$.
  - Thus, at least $\left\lceil \frac{k^{\ell+1}-1}{k} \right\rceil \geqslant k^\ell$ of those u_is are r-dominated by d_p.



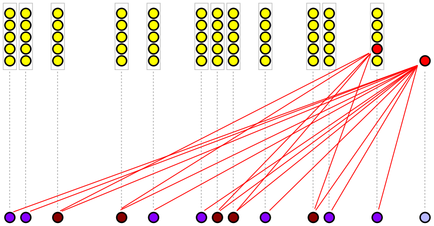
Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.
- Suppose the Algorithm performed $p = k^{\ell+1}$ rounds.
- Examine D_p:
  - Some $d_p \in D_p$ r-dominates $\geqslant \frac{1}{k}$ fraction of $\{u_1, \ldots, u_{p-1}\}$.
  - Thus, at least $\left\lceil \frac{k^{\ell+1}-1}{k} \right\rceil \geqslant k^\ell$ of those u_is are r-dominated by d_p.
  - Keep d_p and u_p, and forget about all D_is and u_is apart from those for which u_i is r-dominated by d_p.



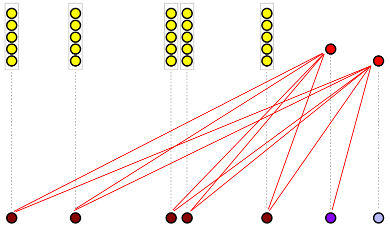
Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.
- Suppose the Algorithm performed $p = k^{\ell+1}$ rounds.
- Examine D_p:
  - Some $d_p \in D_p$ r-dominates $\geqslant \frac{1}{k}$ fraction of $\{u_1, \ldots, u_{p-1}\}$.
  - Thus, at least $\left\lceil \frac{k^{\ell+1}-1}{k} \right\rceil \geqslant k^{\ell}$ of those u_is are r-dominated by d_p.
  - Keep d_p and u_p, and forget about all D_is and u_is apart from those for which u_i is r-dominated by d_p.
- $\ell + 1$ iterations $\rightsquigarrow$ a quasi-ladder of length $\ell + 1$, a contradiction.



Case $k > 1$

- For $k = 1$ the number of rounds is at most ℓ.
- **Claim**: For $k > 1$, the number of rounds less than $k^{\ell+1}$.
- Suppose the Algorithm performed $p = k^{\ell+1}$ rounds.
- Examine D_p:
  - Some $d_p \in D_p$ r-dominates $\geqslant \frac{1}{k}$ fraction of $\{u_1, \ldots, u_{p-1}\}$.
  - Thus, at least $\left\lceil \frac{k^{\ell+1}-1}{k} \right\rceil \geqslant k^\ell$ of those u_is are r-dominated by d_p.
  - Keep d_p and u_p, and forget about all D_is and u_is apart from those for which u_i is r-dominated by d_p.
- $\ell + 1$ iterations $\rightsquigarrow$ a quasi-ladder of length $\ell + 1$, a contradiction.



Stability

- Let $\delta(\bar{x}, \bar{y})$ be an FO formula, where $\bar{x}, \bar{y}$ are tuples of variables.

Stability

- Let $\delta(\bar{x}, \bar{y})$ be an FO formula, where $\bar{x}, \bar{y}$ are tuples of variables.
  - **Ex**: $\delta(x, y) =$ "$\mathrm{dist}(x, y) \leqslant r$" for a fixed r.

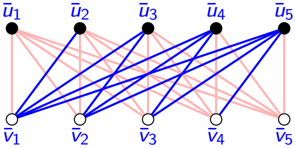
Stability

- Let $\delta(\bar{x}, \bar{y})$ be an FO formula, where $\bar{x}, \bar{y}$ are tuples of variables.
  - **Ex**: $\delta(x, y) =$ "$\mathrm{dist}(x, y) \leqslant r$" for a fixed r.
- **Def**: δ**-ladder** of length ℓ in $G = (V, E)$ is a pair of tuple sequences

$$\bar{u}_1, \ldots, \bar{u}_\ell \qquad \text{and} \qquad \bar{v}_1, \ldots, \bar{v}_\ell$$

belonging to $V^{\bar{x}}$ and $V^{\bar{y}}$, resp., s.t. for all $i, j \in \{1, \ldots, \ell\}$ we have

$$G \models \delta(\bar{u}_i, \bar{v}_j) \qquad \text{if and only if} \qquad i > j.$$



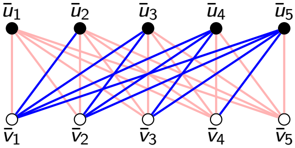
Stability

- Let $\delta(\bar{x}, \bar{y})$ be an FO formula, where $\bar{x}, \bar{y}$ are tuples of variables.
  - **Ex**: $\delta(x, y) =$ "$\mathrm{dist}(x, y) \leqslant r$" for a fixed r.
- **Def**: δ-**ladder** of length ℓ in $G = (V, E)$ is a pair of tuple sequences

$$\bar{u}_1, \ldots, \bar{u}_\ell \qquad \text{and} \qquad \bar{v}_1, \ldots, \bar{v}_\ell$$

belonging to $V^{\bar{x}}$ and $V^{\bar{y}}$, resp., s.t. for all $i, j \in \{1, \ldots, \ell\}$ we have

$$G \models \delta(\bar{u}_i, \bar{v}_j) \qquad \text{if and only if} \qquad i > j.$$



- **Def**: a graph class $\mathcal{C}$ is **stable** if for every formula δ there is a finite bound on the length of δ-ladders in graphs from $\mathcal{C}$.

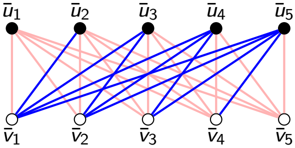
Stability

- Let $\delta(\bar{x}, \bar{y})$ be an FO formula, where $\bar{x}, \bar{y}$ are tuples of variables.
  - **Ex**: $\delta(x, y) =$ "$\mathrm{dist}(x, y) \leqslant r$" for a fixed r.
- **Def**: δ-**ladder** of length ℓ in $G = (V, E)$ is a pair of tuple sequences

$$\bar{u}_1, \ldots, \bar{u}_\ell \qquad \text{and} \qquad \bar{v}_1, \ldots, \bar{v}_\ell$$

belonging to $V^{\bar{x}}$ and $V^{\bar{y}}$, resp., s.t. for all $i, j \in \{1, \ldots, \ell\}$ we have

$$G \models \delta(\bar{u}_i, \bar{v}_j) \qquad \text{if and only if} \qquad i > j.$$



- **Def**: a graph class $\mathcal{C}$ is **stable** if for every formula δ there is a finite bound on the length of δ-ladders in graphs from $\mathcal{C}$.
  - **Intuition**: One cannot define arbitrary long linear orders.

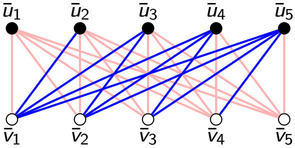
Stability

- Let $\delta(\bar{x}, \bar{y})$ be an FO formula, where $\bar{x}, \bar{y}$ are tuples of variables.
  - **Ex**: $\delta(x, y) =$ "$\mathrm{dist}(x, y) \leqslant r$" for a fixed r.
- **Def**: δ-**ladder** of length ℓ in $G = (V, E)$ is a pair of tuple sequences

$$\bar{u}_1, \ldots, \bar{u}_\ell \qquad \text{and} \qquad \bar{v}_1, \ldots, \bar{v}_\ell$$

belonging to $V^{\bar{x}}$ and $V^{\bar{y}}$, resp., s.t. for all $i, j \in \{1, \ldots, \ell\}$ we have

$$G \models \delta(\bar{u}_i, \bar{v}_j) \qquad \text{if and only if} \qquad i > j.$$



- **Def**: a graph class $\mathcal{C}$ is **stable** if for every formula δ there is a finite bound on the length of δ-ladders in graphs from $\mathcal{C}$.
  - **Intuition**: One cannot define arbitrary long linear orders.
  - **Use**: upper bounding the number of rounds in iterative constructions.

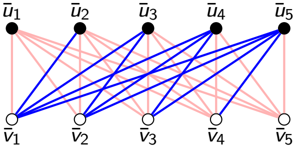
Stability

- Let $\delta(\bar{x}, \bar{y})$ be an FO formula, where $\bar{x}, \bar{y}$ are tuples of variables.
  - **Ex**: $\delta(x, y) =$ "$\mathrm{dist}(x, y) \leqslant r$" for a fixed r.
- **Def**: δ-**ladder** of length ℓ in $G = (V, E)$ is a pair of tuple sequences

$$\bar{u}_1, \ldots, \bar{u}_\ell \qquad \text{and} \qquad \bar{v}_1, \ldots, \bar{v}_\ell$$

belonging to $V^{\bar{x}}$ and $V^{\bar{y}}$, resp., s.t. for all $i, j \in \{1, \ldots, \ell\}$ we have

$$G \models \delta(\bar{u}_i, \bar{v}_j) \qquad \text{if and only if} \qquad i > j.$$



- **Def**: a graph class $\mathcal{C}$ is **stable** if for every formula δ there is a finite bound on the length of δ-ladders in graphs from $\mathcal{C}$.
  - **Intuition**: One cannot define arbitrary long linear orders.
  - **Use**: upper bounding the number of rounds in iterative constructions.
- **Thm**: A monotone graph class is stable iff it is nowhere dense.

Conclusions

- Theory of structural sparsity in a nutshell:

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for "local" problems.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for “local” problems.
  - Toolbox applicable under various algorithmic paradigms, with parameterized complexity being probably the foremost.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for "local" problems.
  - Toolbox applicable under various algorithmic paradigms, with parameterized complexity being probably the foremost.
  - Provides clean proofs of strong and general results.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for "local" problems.
  - Toolbox applicable under various algorithmic paradigms, with parameterized complexity being probably the foremost.
  - Provides clean proofs of strong and general results.
  - **Caveat**: Hidden constants are terrible.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for "local" problems.
  - Toolbox applicable under various algorithmic paradigms, with parameterized complexity being probably the foremost.
  - Provides clean proofs of strong and general results.
  - **Caveat**: Hidden constants are terrible.
- If interested, see **lecture notes**: `https://www.mimuw.edu.pl/~mp248287/sparsity/`

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for "local" problems.
  - Toolbox applicable under various algorithmic paradigms, with parameterized complexity being probably the foremost.
  - Provides clean proofs of strong and general results.
  - **Caveat**: Hidden constants are terrible.
- If interested, see **lecture notes**:
  `https://www.mimuw.edu.pl/~mp248287/sparsity/`
  - So far we covered the core part of the theory in 10 lectures.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for "local" problems.
  - Toolbox applicable under various algorithmic paradigms, with parameterized complexity being probably the foremost.
  - Provides clean proofs of strong and general results.
  - **Caveat**: Hidden constants are terrible.
- If interested, see **lecture notes**:
  `https://www.mimuw.edu.pl/~mp248287/sparsity/`
  - So far we covered the core part of the theory in 10 lectures.
  - Probably at some point they will be turned into a book.

Conclusions

- Theory of structural sparsity in a nutshell:
  - Basic definitions capture the concept of sparsity that persists under local contractions.
  - An abundance of equivalent characterizations and viewpoints.
  - Each characterization provides a tool that can be used to study **combinatorial**, **algorithmic**, and **logical** aspects.
  - **Intuition**: Delimits the border of tractability for "local" problems.
  - Toolbox applicable under various algorithmic paradigms, with parameterized complexity being probably the foremost.
  - Provides clean proofs of strong and general results.
  - **Caveat**: Hidden constants are terrible.
- If interested, see **lecture notes**: `https://www.mimuw.edu.pl/~mp248287/sparsity/`
  - So far we covered the core part of the theory in 10 lectures.
  - Probably at some point they will be turned into a book.
- **Thank you for your attention!**