Sparse graphs

Michał Pilipczuk

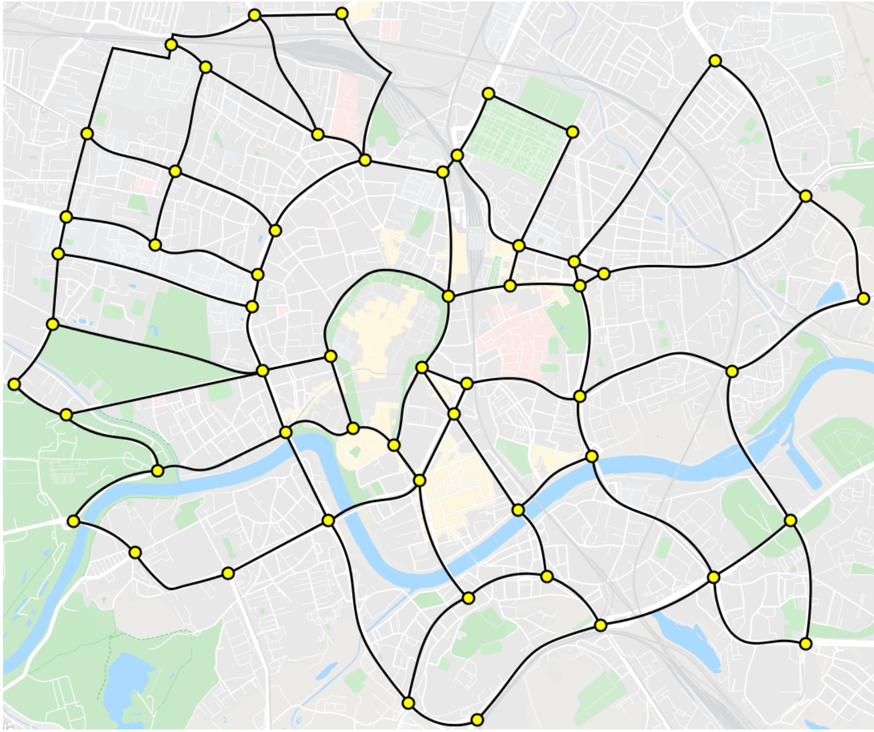
Faculty of Mathematics, Informatics, and Mechanics
University of Warsaw

January 9th, 2020

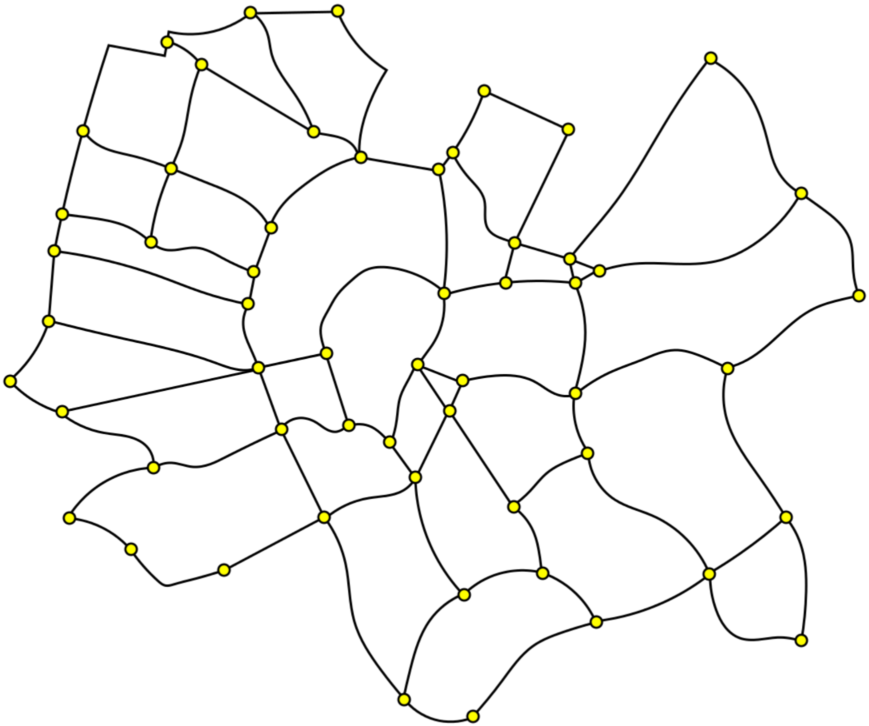
Graphs

Graphs

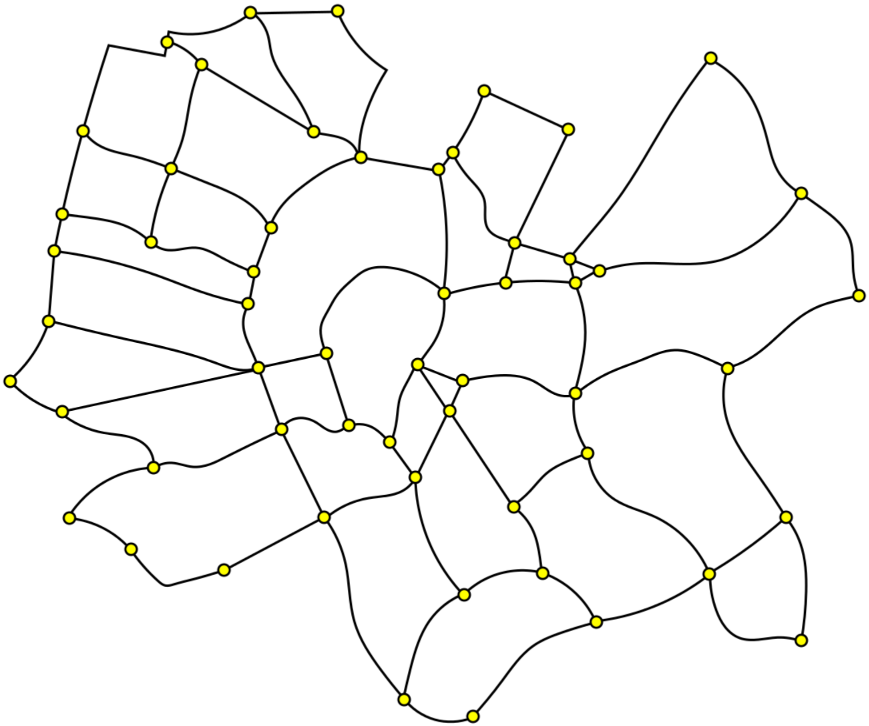
Graphs



Graphs

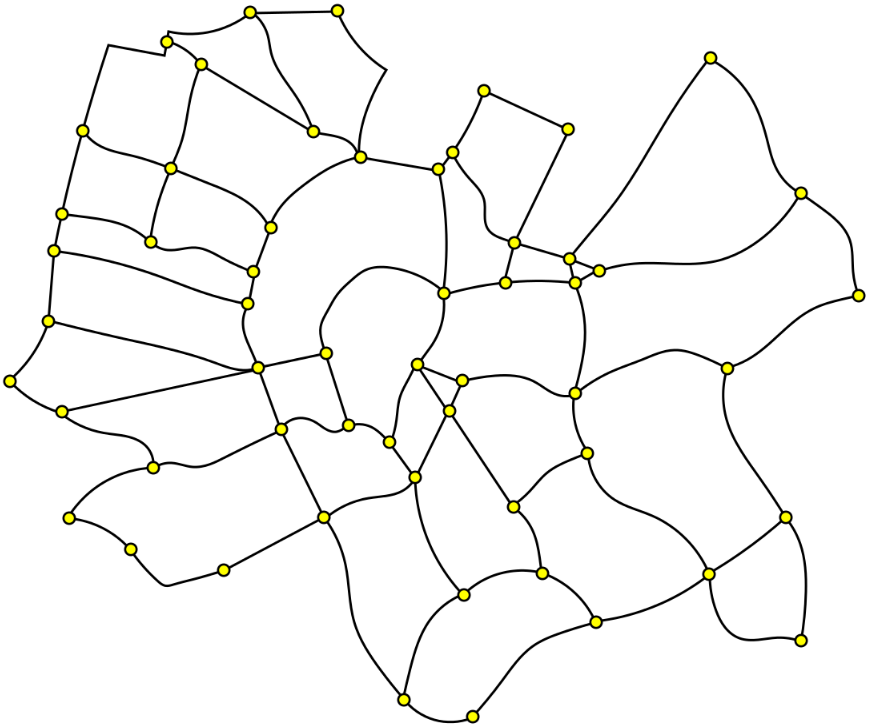


Graphs



Graph: a set of vertices connected in pairs by edges

Graphs



Graph: a set of vertices connected in pairs by edges

- **Finite** for the purpose of this talk.
- **Simple:** no two edges connect the same pair of vertices.

Graphs in practice

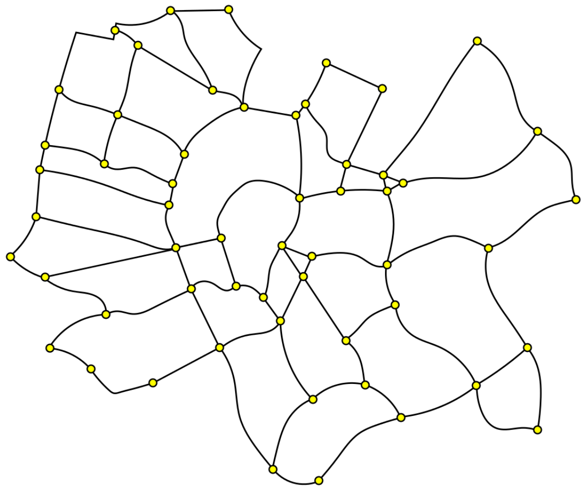
Graphs in practice

Graphs may model:

Graphs in practice

Graphs may model:

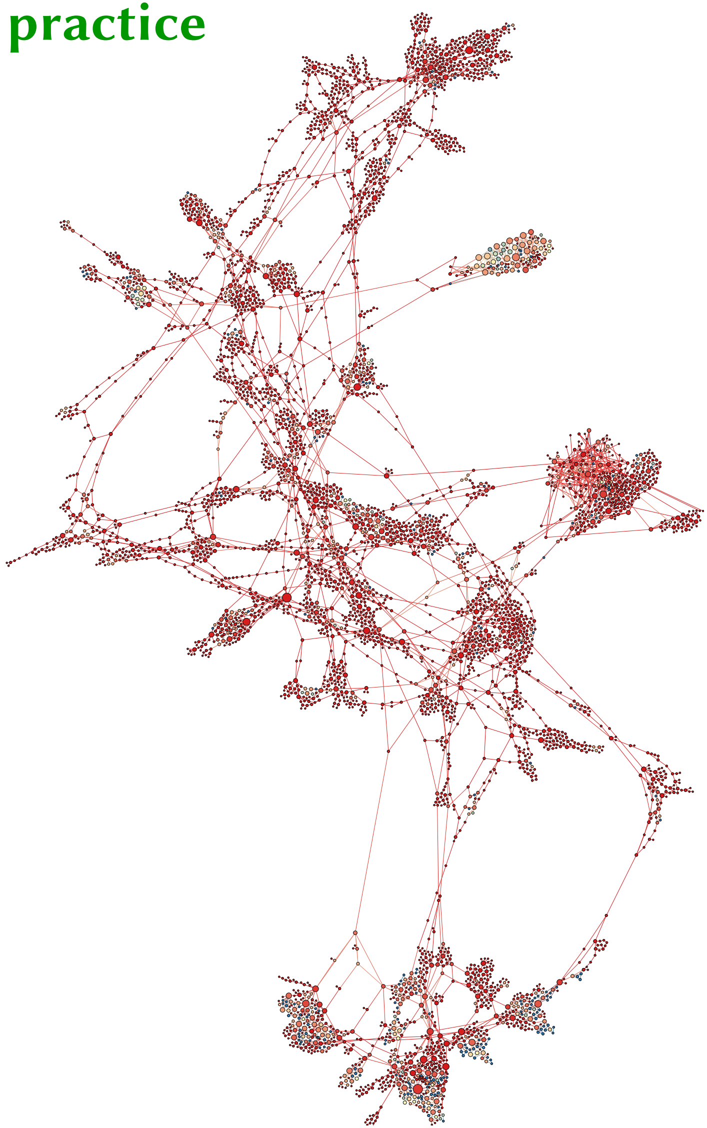
– Transportation networks



Graphs in practice

Graphs may model:

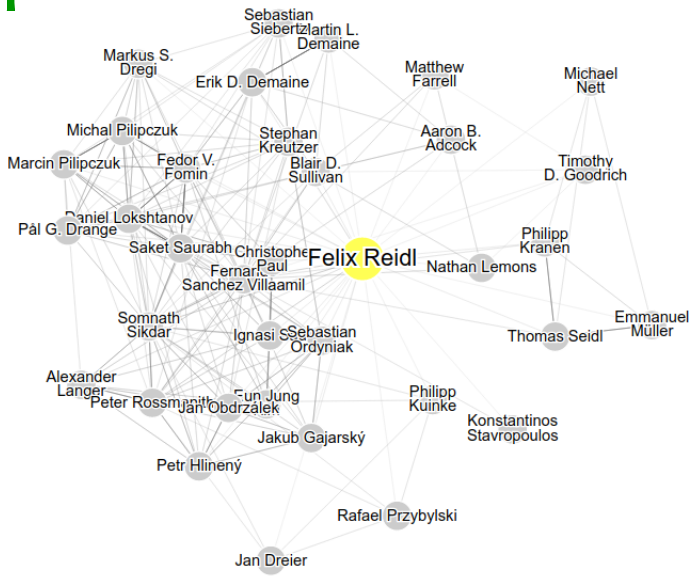
- Transportation networks
- Infrastructure



Graphs in practice

Graphs may model:

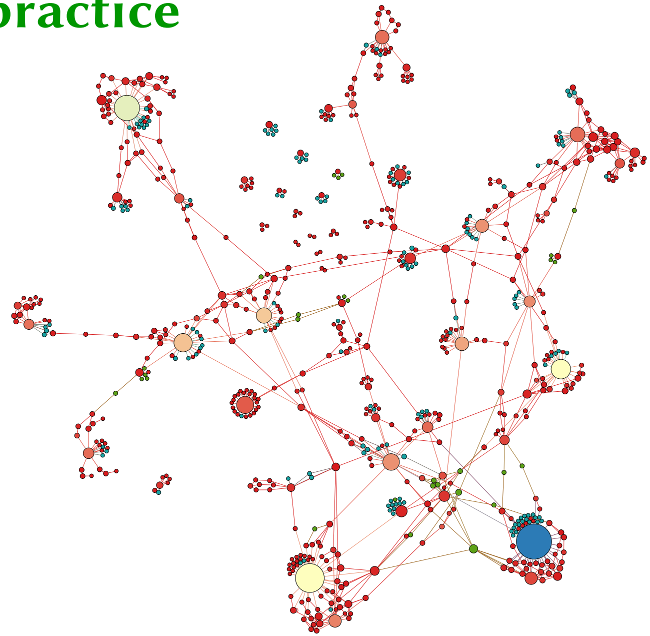
- Transportation networks
- Infrastructure
- Social networks



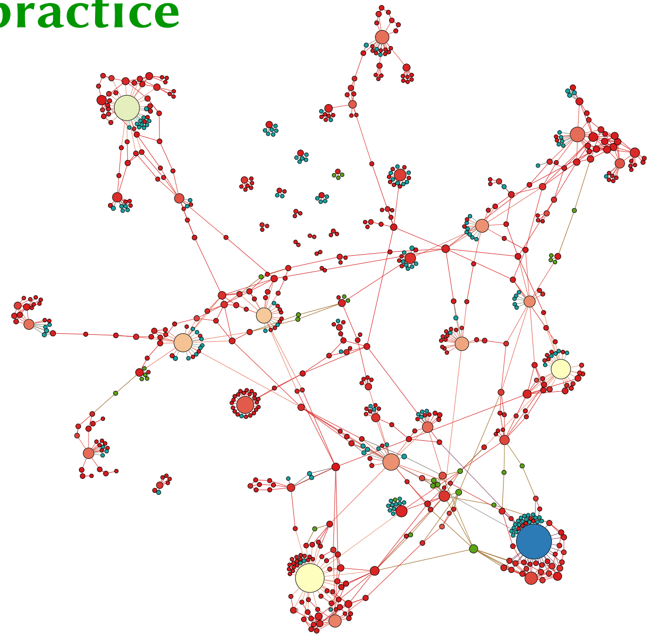
Graphs in practice

Graphs may model:

- Transportation networks
- Infrastructure
- Social networks
- Relations in data



Graphs in practice



Graphs may model:

– Transportation networks

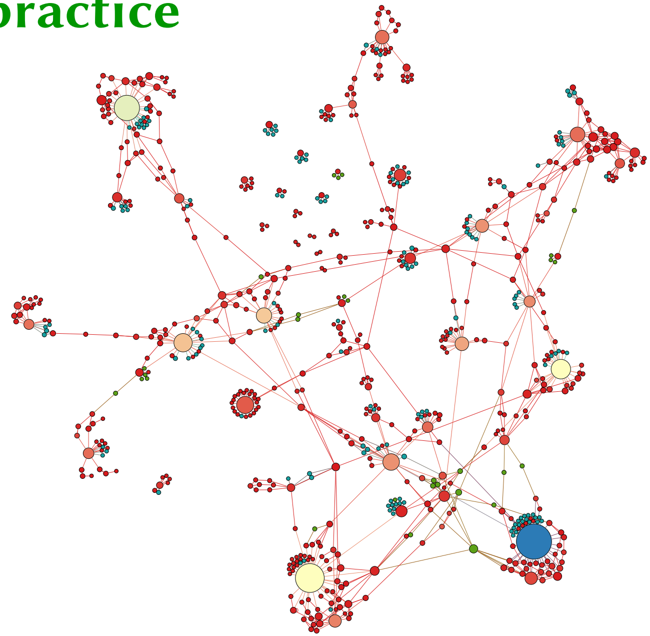
– Infrastructure

– Social networks

– Relations in data

Graphs tend to be **sparse**.

Graphs in practice



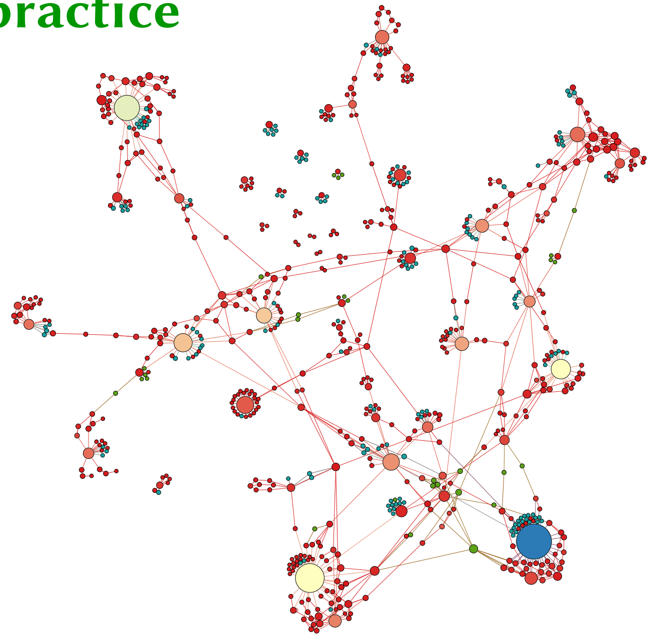
Graphs may model:

- Transportation networks
- Infrastructure
- Social networks
- Relations in data

Graphs tend to be **sparse**.

- Transportation networks are (roughly) planar.

Graphs in practice



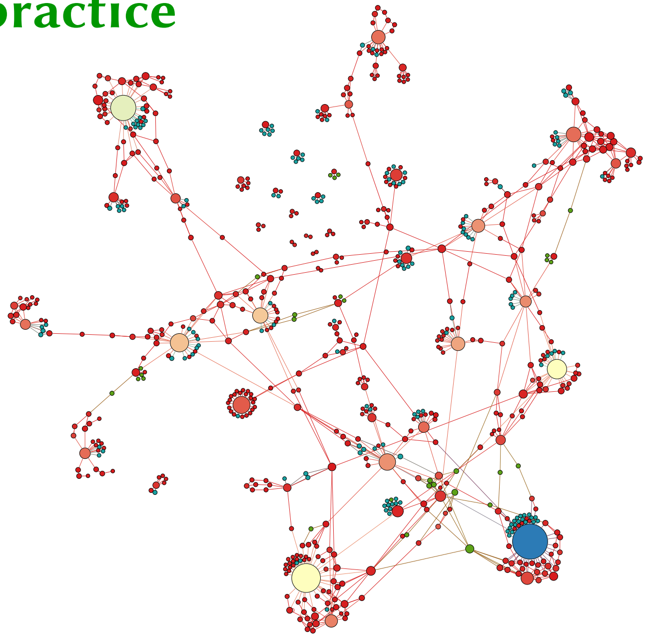
Graphs may model:

- Transportation networks
- Infrastructure
- Social networks
- Relations in data

Graphs tend to be **sparse**.

- Transportation networks are (roughly) planar.
- Facebook graph has average degree 338 and median degree 200.

Graphs in practice



Graphs may model:

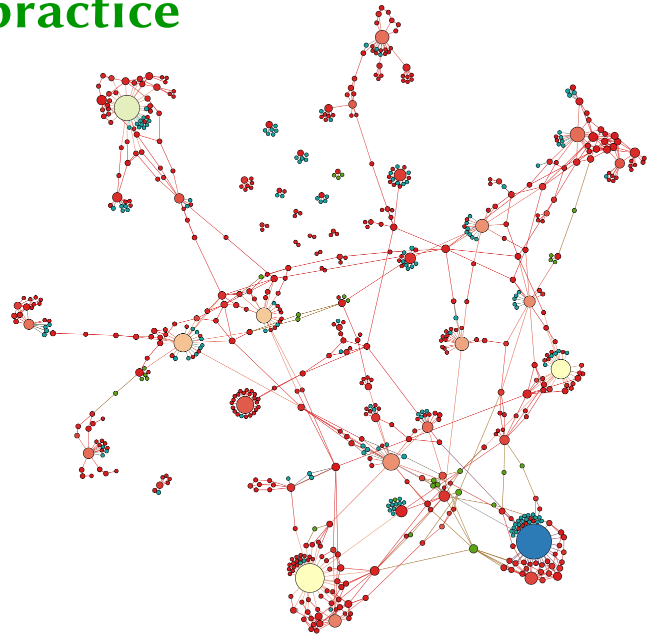
- Transportation networks
- Infrastructure
- Social networks
- Relations in data

Graphs tend to be **sparse**.

- Transportation networks are (roughly) planar.
- Facebook graph has average degree 338 and median degree 200.

What does it mean **sparse**?

Graphs in practice



Graphs may model:

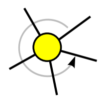
- Transportation networks
- Infrastructure
- Social networks
- Relations in data

Graphs tend to be **sparse**.

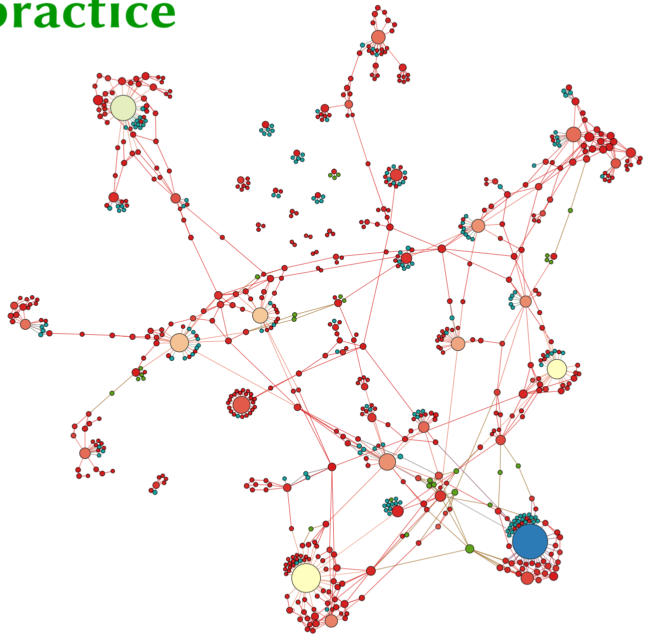
- Transportation networks are (roughly) planar.
- Facebook graph has average degree 338 and median degree 200.

What does it mean **sparse**?

- Bounded degree?



Graphs in practice



Graphs may model:

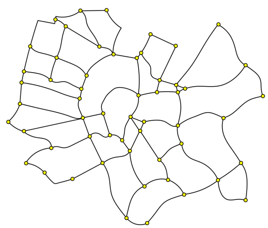
- Transportation networks
- Infrastructure
- Social networks
- Relations in data

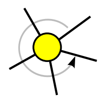
Graphs tend to be **sparse**.

- Transportation networks are (roughly) planar.
- Facebook graph has average degree 338 and median degree 200.

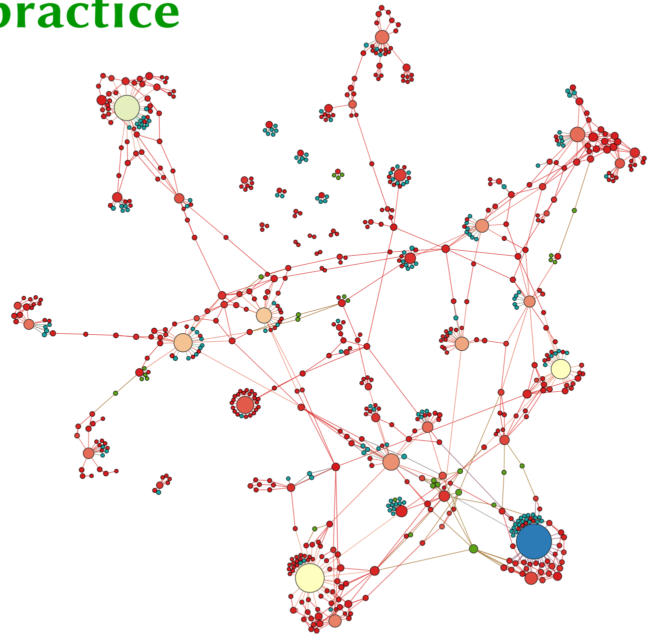
What does it mean **sparse**?

- Bounded degree?
- Planar-like? Tree-like?





Graphs in practice



Graphs may model:

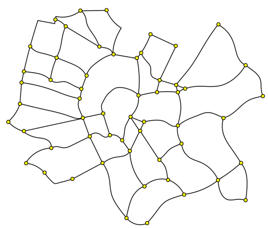
- Transportation networks
- Infrastructure
- Social networks
- Relations in data

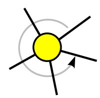
Graphs tend to be **sparse**.

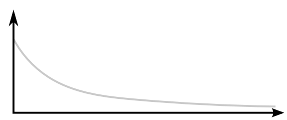
- Transportation networks are (roughly) planar.
- Facebook graph has average degree 338 and median degree 200.

What does it mean **sparse**?

- Bounded degree?
- Planar-like? Tree-like?
- Fixed degree distribution?







Objectives of Sparsity

Objectives of Sparsity

Goal. A **theory** of sparse graphs that is:

Objectives of Sparsity

Goal. A **theory** of sparse graphs that is:

1. general and robust;

Objectives of Sparsity

Goal. A **theory** of sparse graphs that is:

1. general and robust;
2. mathematically elegant and interesting;

Objectives of Sparsity

Goal. A **theory** of sparse graphs that is:

1. general and robust;

2. mathematically elegant and interesting;

3. useful in applications.

Objectives of Sparsity

Goal. A **theory** of sparse graphs that is:

1. general and robust;

2. mathematically elegant and interesting;

3. useful in applications.

Sparsity: a young area of graph theory that $\pm$ achieves all the above.

Objectives of Sparsity

Goal. A **theory** of sparse graphs that is:

1. general and robust;

2. mathematically elegant and interesting;

3. useful in applications.

Sparsity: a young area of graph theory that $\pm$ achieves all the above.

– **Focus:** Abstract notions of **local** and **uniform sparseness**.

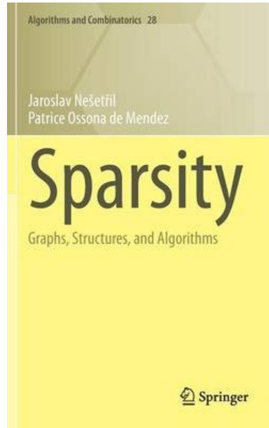
Objectives of Sparsity

Goal. A **theory** of sparse graphs that is:

1. general and robust;

2. mathematically elegant and interesting;

3. useful in applications.

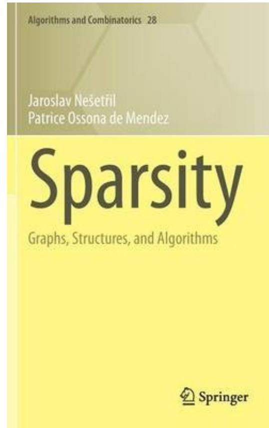


Sparsity: a young area of graph theory that $\pm$ achieves all the above.

– **Focus:** Abstract notions of **local** and **uniform sparseness**.

– Initiated $\sim$2008 by Jaroslav Nešetřil and Patrice Ossona de Mendez.

Objectives of Sparsity



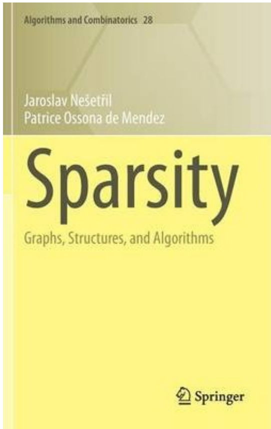
Goal. A **theory** of sparse graphs that is:

1. general and robust;
2. mathematically elegant and interesting;
3. useful in applications.

Sparsity: a young area of graph theory that $\pm$ achieves all the above.

- **Focus:** Abstract notions of **local** and **uniform sparseness**.
- Initiated $\sim$2008 by Jaroslav Nešetřil and Patrice Ossona de Mendez.
- Since then, tremendous development of techniques and concepts.

Objectives of Sparsity



Goal. A **theory** of sparse graphs that is:

1. general and robust;
2. mathematically elegant and interesting;
3. useful in applications.

Sparsity: a young area of graph theory that $\pm$ achieves all the above.

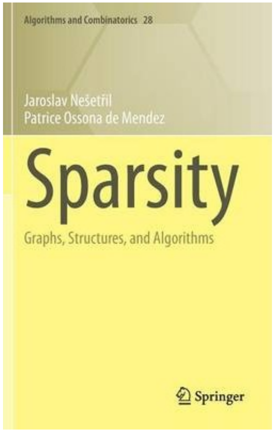
– **Focus:** Abstract notions of **local** and **uniform sparseness**.

– Initiated $\sim$2008 by Jaroslav Nešetřil and Patrice Ossona de Mendez.

– Since then, tremendous development of techniques and concepts.

Now: a very brief introduction to the area.

Objectives of Sparsity



Goal. A **theory** of sparse graphs that is:

1. general and robust;
2. mathematically elegant and interesting;
3. useful in applications.

Sparsity: a young area of graph theory that $\pm$ achieves all the above.

– **Focus:** Abstract notions of **local** and **uniform sparseness**.

– Initiated ∼2008 by Jaroslav Nešetřil and Patrice Ossona de Mendez.

– Since then, tremendous development of techniques and concepts.

Now: a very brief introduction to the area.

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

Measuring sparsity

Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

– Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

– Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

$$\mathrm{avgdeg}(G) = \frac{\sum_{u \in V(G)} \deg(u)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$$

Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

– Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

$$\mathrm{avgdeg}(G) = \frac{\sum_{u \in V(G)} \deg(u)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$$

– Equivalently, average degree in G is bounded by $2c$.

Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

– Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

$$\mathrm{avgdeg}(G) = \frac{\sum_{u \in V(G)} \deg(u)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$$

– Equivalently, average degree in G is bounded by $2c$.

Ex 1. Maximum degree $\leqslant d \quad \Rightarrow \quad$ Average degree $\leqslant d$.

Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

– Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

$$\mathrm{avgdeg}(G) = \frac{\sum_{u \in V(G)} \deg(u)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$$

– Equivalently, average degree in G is bounded by $2c$.

Ex 1. Maximum degree $\leqslant d \quad \Rightarrow \quad$ Average degree $\leqslant d$.

Ex 2. Planar graph has $\leqslant 3n - 6$ edges $\quad \Rightarrow \quad$ Average degree < 6.

Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

- Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

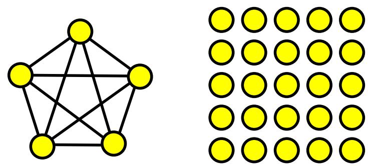
$$\mathrm{avgdeg}(G) = \frac{\sum_{u \in V(G)} \deg(u)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$$

- Equivalently, average degree in G is bounded by $2c$.

Ex 1. Maximum degree $\leqslant d \quad \Rightarrow \quad$ Average degree $\leqslant d$.

Ex 2. Planar graph has $\leqslant 3n - 6$ edges $\quad \Rightarrow \quad$ Average degree < 6.

Issue: A complete graph on k vertices plus k^2 isolated vertices.



Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

– Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

$$\mathrm{avgdeg}(G) = \frac{\sum_{u \in V(G)} \deg(u)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$$

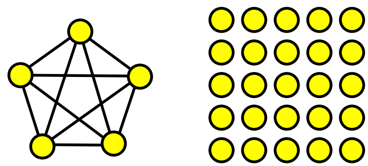
– Equivalently, average degree in G is bounded by $2c$.

Ex 1. Maximum degree $\leqslant d \quad \Rightarrow \quad$ Average degree $\leqslant d$.

Ex 2. Planar graph has $\leqslant 3n - 6$ edges $\quad \Rightarrow \quad$ Average degree < 6.

Issue: A complete graph on k vertices plus k^2 isolated vertices.

– Average degree smaller than 1.



Measuring sparsity

Attempt 1. A graph G is **sparse** if it has a linear number of edges.

– Formally, $|E(G)| \leqslant c \cdot |V(G)|$ for some constant c.

$$\mathrm{avgdeg}(G) = \frac{\sum_{u \in V(G)} \deg(u)}{|V(G)|} = \frac{2|E(G)|}{|V(G)|}$$

– Equivalently, average degree in G is bounded by $2c$.

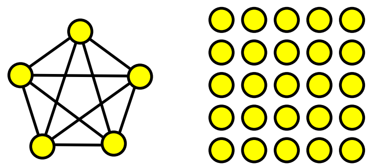
Ex 1. Maximum degree $\leqslant d \quad \Rightarrow \quad$ Average degree $\leqslant d$.

Ex 2. Planar graph has $\leqslant 3n - 6$ edges $\quad \Rightarrow \quad$ Average degree < 6.

Issue: A complete graph on k vertices plus k^2 isolated vertices.

– Average degree smaller than 1.

– Contains a **dense** subgraph.



Measuring sparsity

Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

– We define **maximum average degree** of G as

$$\mathrm{mad}(G) := \max_{H \subseteq G} \mathrm{avgdeg}(H).$$

Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

- We define **maximum average degree** of G as

$$\mathrm{mad}(G) \coloneqq \max_{H \subseteq G} \mathrm{avgdeg}(H).$$

- G is **sparse** if $\mathrm{mad}(G) \leqslant c$ for some constant c.

Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

– We define **maximum average degree** of G as

$$\mathrm{mad}(G) \coloneqq \max_{H \subseteq G} \mathrm{avgdeg}(H).$$

– G is **sparse** if $\mathrm{mad}(G) \leqslant c$ for some constant c.

Ex 1. G has maximum degree $\leqslant d \quad \Rightarrow \quad \mathrm{mad}(G) \leqslant d$.

Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

– We define **maximum average degree** of G as

$$\operatorname{mad}(G) \coloneqq \max_{H \subseteq G} \operatorname{avgdeg}(H).$$

– G is **sparse** if $\operatorname{mad}(G) \leqslant c$ for some constant c.

Ex 1. G has maximum degree $\leqslant d \quad \Rightarrow \quad \operatorname{mad}(G) \leqslant d$.

Ex 2. G is planar $\quad \Rightarrow \quad \operatorname{mad}(G) < 6$.

Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

- We define **maximum average degree** of G as

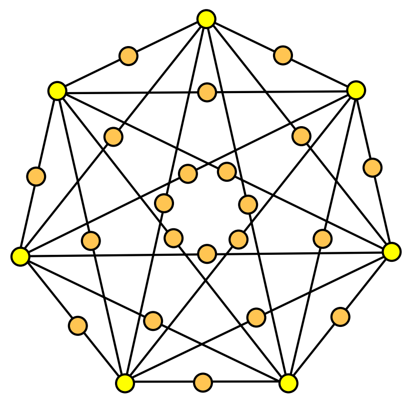
$$\mathrm{mad}(G) := \max_{H \subseteq G} \mathrm{avgdeg}(H).$$

- G is **sparse** if $\mathrm{mad}(G) \leqslant c$ for some constant c.

Ex 1. G has maximum degree $\leqslant d \quad \Rightarrow \quad \mathrm{mad}(G) \leqslant d$.

Ex 2. G is planar $\quad \Rightarrow \quad \mathrm{mad}(G) < 6$.

Issue: A subdivided complete graph.



Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

– We define **maximum average degree** of G as

$$\mathrm{mad}(G) := \max_{H \subseteq G} \mathrm{avgdeg}(H).$$

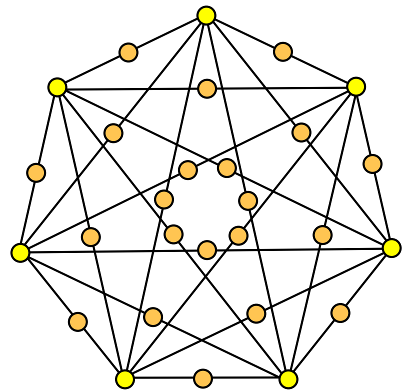
– G is **sparse** if $\mathrm{mad}(G) \leqslant c$ for some constant c.

Ex 1. G has maximum degree $\leqslant d \quad \Rightarrow \quad \mathrm{mad}(G) \leqslant d$.

Ex 2. G is planar $\quad \Rightarrow \quad \mathrm{mad}(G) < 6$.

Issue: A subdivided complete graph.

– **Exc:** Every subgraph has avgdeg $\leqslant 4$.



Measuring sparsity

Attempt 2. Every **subgraph** of G has a linear number of edges.

– We define **maximum average degree** of G as

$$\mathrm{mad}(G) := \max_{H \subseteq G} \mathrm{avgdeg}(H).$$

– G is **sparse** if $\mathrm{mad}(G) \leqslant c$ for some constant c.

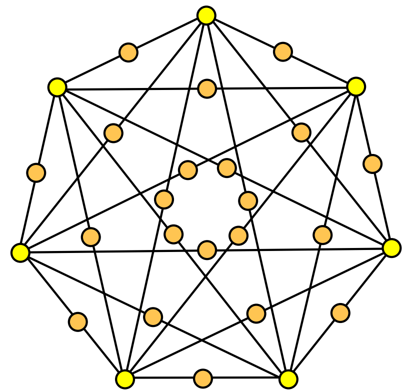
Ex 1. G has maximum degree $\leqslant d \quad \Rightarrow \quad \mathrm{mad}(G) \leqslant d$.

Ex 2. G is planar $\quad \Rightarrow \quad \mathrm{mad}(G) < 6$.

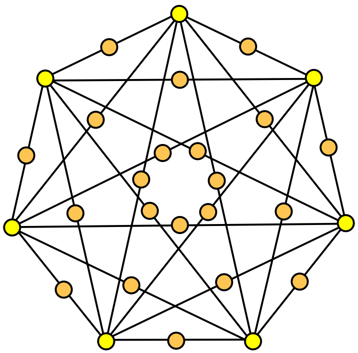
Issue: A subdivided complete graph.

– **Exc:** Every subgraph has avgdeg $\leqslant 4$.

– Is this graph really **sparse**?

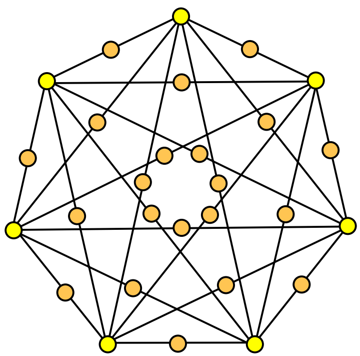


Measuring sparsity



Measuring sparsity

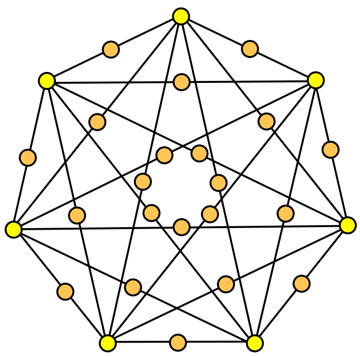
Option 1. We decide that a subdivided complete graph is **sparse**.



Measuring sparsity

Option 1. We decide that a subdivided complete graph is **sparse**.

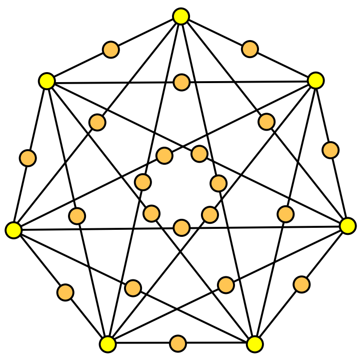
– We can construct a theory around the parameter $\mathrm{mad}(\cdot)$.

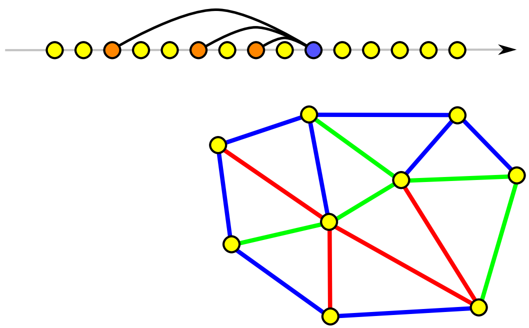


Measuring sparsity

Option 1. We decide that a subdivided complete graph is **sparse**.

- We can construct a theory around the parameter $\mathrm{mad}(\cdot)$.
- $\mathrm{mad}(\cdot)$ is essentially equivalent to **arboricity** and **degeneracy**.





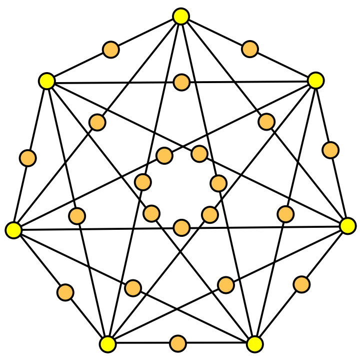
Measuring sparsity

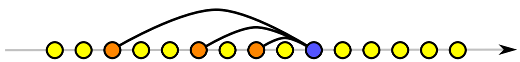
Option 1. We decide that a subdivided complete graph is **sparse**.

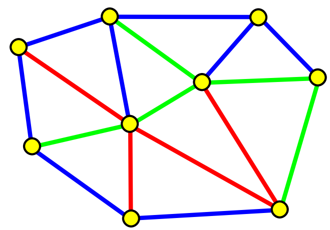
– We can construct a theory around the parameter $\mathrm{mad}(\cdot)$.

– $\mathrm{mad}(\cdot)$ is essentially equivalent to **arboricity** and **degeneracy**.

– These connections are useful, but not really very deep.





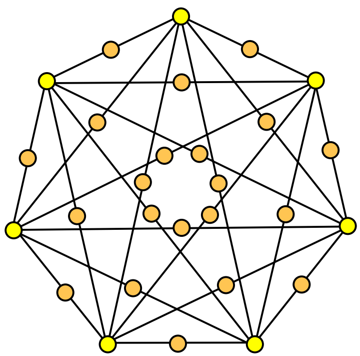


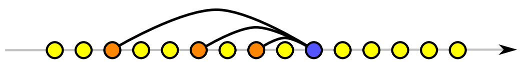
Measuring sparsity

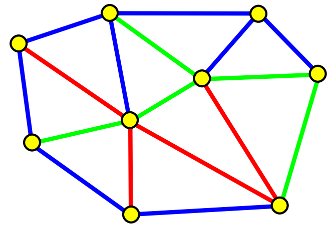
Option 1. We decide that a subdivided complete graph is **sparse**.

- We can construct a theory around the parameter $\mathrm{mad}(\cdot)$.
- $\mathrm{mad}(\cdot)$ is essentially equivalent to **arboricity** and **degeneracy**.
- These connections are useful, but not really very deep.

Option 2. We decide that a subdivided complete graph is **dense**.







Measuring sparsity

Option 1. We decide that a subdivided complete graph is **sparse**.

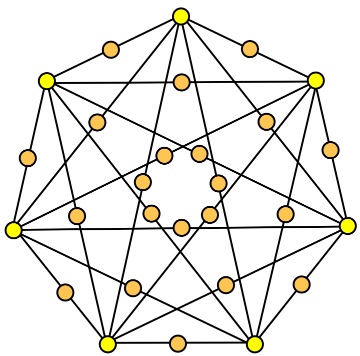
– We can construct a theory around the parameter $\mathrm{mad}(\cdot)$.

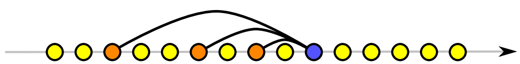
– $\mathrm{mad}(\cdot)$ is essentially equivalent to **arboricity** and **degeneracy**.

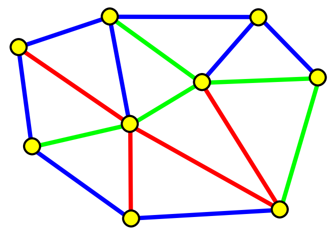
– These connections are useful, but not really very deep.

Option 2. We decide that a subdivided complete graph is **dense**.

– **Reason:** It contains a dense substructure visible at "depth" 1.







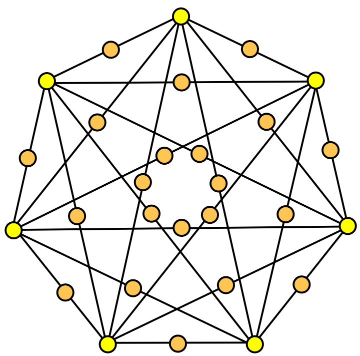
Measuring sparsity

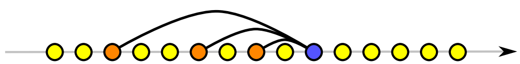
Option 1. We decide that a subdivided complete graph is **sparse**.

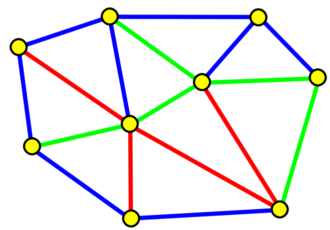
- We can construct a theory around the parameter $\mathrm{mad}(\cdot)$.
- $\mathrm{mad}(\cdot)$ is essentially equivalent to **arboricity** and **degeneracy**.
- These connections are useful, but not really very deep.

Option 2. We decide that a subdivided complete graph is **dense**.

- **Reason:** It contains a dense substructure visible at "depth" 1.
- **Need:** A notion of **embedding** that would capture this.

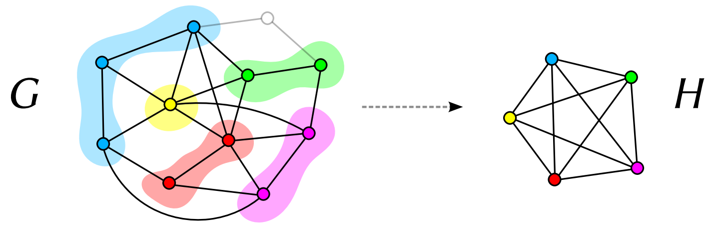






Minor order

Minor order

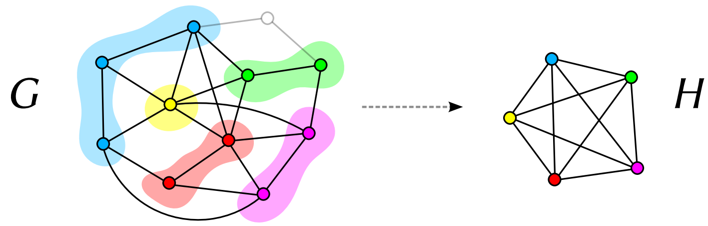


Definition

H is a **minor** of G $\Leftrightarrow$

H is obtained from a subgraph of G by contracting connected subgraphs

Minor order



Definition

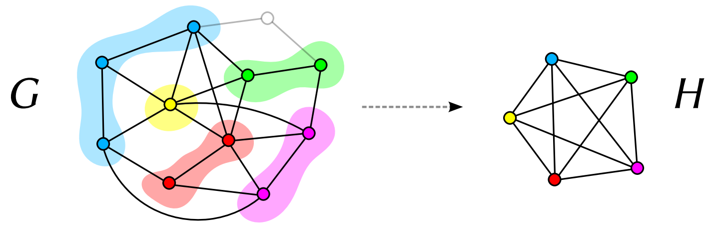
H is a **minor** of G $\Leftrightarrow$

H is obtained from a subgraph of G by contracting connected subgraphs

Theorem (Kuratowski; Wagner)

Planar graphs are exactly $\{K_5, K_{3,3}\}$-minor-free graphs.

Minor order



Definition

H is a **minor** of G $\Leftrightarrow$

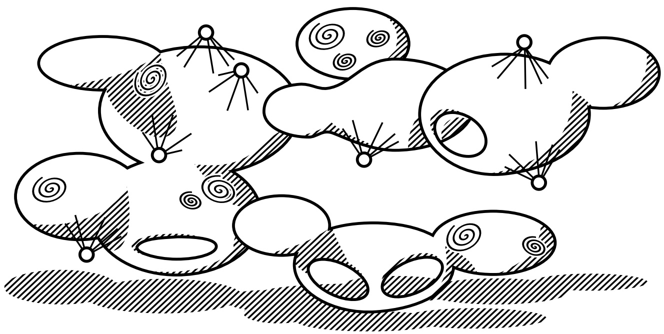
H is obtained from a subgraph of G by contracting connected subgraphs

Theorem (Kuratowski; Wagner)

Planar graphs are exactly $\{K_5, K_{3,3}\}$-minor-free graphs.

Theorem (Robertson and Seymour)

For every $t \in \mathbb{N}$, every K_t-minor-free graph looks like this:



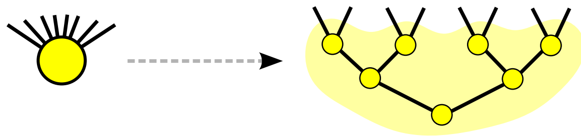
Shallow minors

Attempt 3. Graphs excluding K_t as a minor, for some $t \in \mathbb{N}$.

Shallow minors

Attempt 3. Graphs excluding K_t as a minor, for some $t \in \mathbb{N}$.

Issue: Graphs with maxdeg 3 admit all complete graphs as minors.



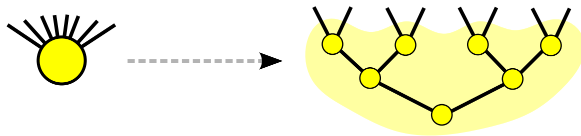
Shallow minors

Attempt 3. Graphs excluding K_t as a minor, for some $t \in \mathbb{N}$.

Issue: Graphs with maxdeg 3 admit all complete graphs as minors.

Ergo: **Excluding minors** leads to an interesting theory,

but this is **not** the theory we are after.



Shallow minors

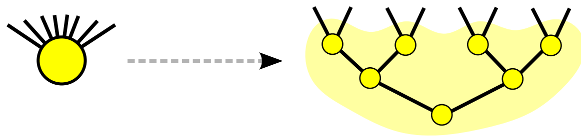
Attempt 3. Graphs excluding K_t as a minor, for some $t \in \mathbb{N}$.

Issue: Graphs with maxdeg 3 admit all complete graphs as minors.

Ergo: **Excluding minors** leads to an interesting theory,

but this is **not** the theory we are after.

Idea: Think about **local** minors.



Shallow minors

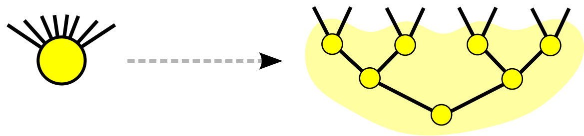
Attempt 3. Graphs excluding K_t as a minor, for some $t \in \mathbb{N}$.

Issue: Graphs with maxdeg 3 admit all complete graphs as minors.

Ergo: **Excluding minors** leads to an interesting theory,

but this is **not** the theory we are after.

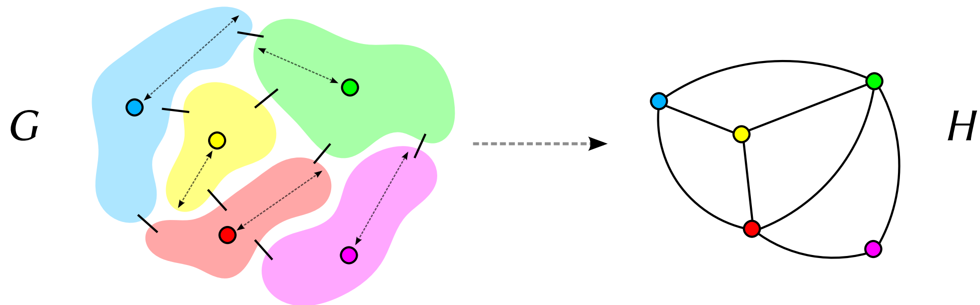
Idea: Think about **local** minors.



Definition

H is a **depth-d minor** of G $\Leftrightarrow$

H is obtained from a subgraph of G by contracting subgraphs of **radius** $\leqslant d$



Shallow minors

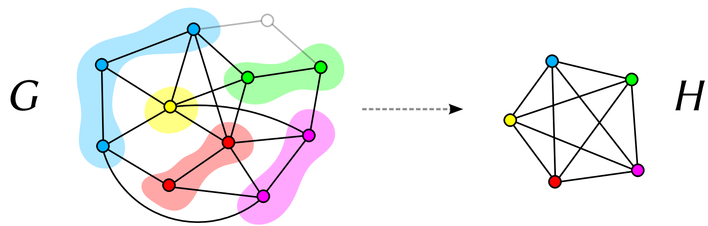
Attempt 3. Graphs excluding K_t as a minor, for some $t \in \mathbb{N}$.

Issue: Graphs with maxdeg 3 admit all complete graphs as minors.

Ergo: **Excluding minors** leads to an interesting theory,

but this is **not** the theory we are after.

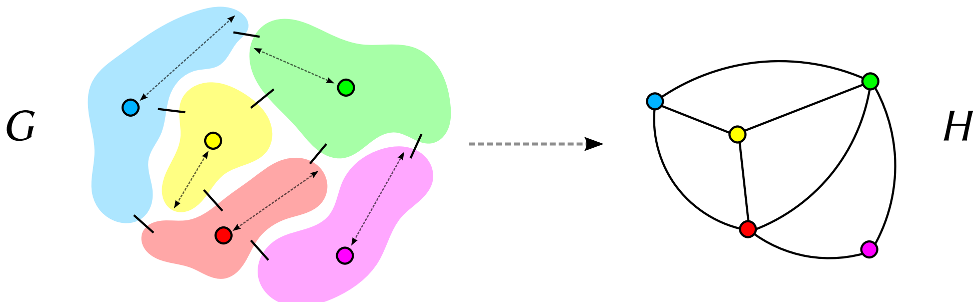
Idea: Think about **local** minors.



Definition

H is a **depth-d minor** of G $\Leftrightarrow$

H is obtained from a subgraph of G by contracting subgraphs of **radius** $\leqslant d$



Notions of sparsity

Notions of sparsity

Intuition: **Sparsity** $\Leftrightarrow$ Exclusion of **dense** structures at every fixed depth

Notions of sparsity

Intuition: Sparsity $\Leftrightarrow$ Exclusion of **dense** structures at every fixed depth

Definition

$$\nabla_d(G) \;:=\; \sup\{\, \mathrm{avgdeg}(H) \;:\; H \text{ is a depth-}d\text{ minor of } G \,\}$$

$$\omega_d(G) \;:=\; \sup\{\, t \;:\; K_t \text{ is a depth-}d\text{ minor of } G \,\}.$$

Notions of sparsity

Intuition: Sparsity $\Leftrightarrow$ Exclusion of **dense** structures at every fixed depth

Definition

$$\begin{aligned} \nabla_d(G) &\coloneqq \sup\{\, \mathrm{avgdeg}(H) \;:\; H \text{ is a depth-}d \text{ minor of } G \,\} \\ \omega_d(G) &\coloneqq \sup\{\, t \;:\; K_t \text{ is a depth-}d \text{ minor of } G \,\}. \end{aligned}$$

Note: depth-0 minors = subgraphs $\rightsquigarrow$ $\nabla_0(G) = \mathrm{mad}(G)$.

Notions of sparsity

Intuition: **Sparsity** $\Leftrightarrow$ Exclusion of **dense** structures at every fixed depth

Definition

$$\nabla_d(G) \coloneqq \sup\{\,\mathrm{avgdeg}(H) : H \text{ is a depth-}d \text{ minor of } G\,\}$$
$$\omega_d(G) \coloneqq \sup\{\,t : K_t \text{ is a depth-}d \text{ minor of } G\,\}.$$

Note: depth-0 minors = subgraphs $\rightsquigarrow$ $\nabla_0(G) = \mathrm{mad}(G)$.

For a **class** of graphs $\mathcal{C}$, we write:

$$\nabla_d(\mathcal{C}) \coloneqq \sup_{G\in\mathcal{C}} \nabla_d(G) \qquad \text{and} \qquad \omega_d(\mathcal{C}) \coloneqq \sup_{G\in\mathcal{C}} \omega_d(G).$$

Notions of sparsity

Intuition: Sparsity $\Leftrightarrow$ Exclusion of **dense** structures at every fixed depth

Definition

$$\nabla_d(G) \coloneqq \sup\{\, \mathrm{avgdeg}(H) \;:\; H \text{ is a depth-}d\text{ minor of } G \,\}$$
$$\omega_d(G) \coloneqq \sup\{\, t \;:\; K_t \text{ is a depth-}d\text{ minor of } G \,\}.$$

Note: depth-0 minors = subgraphs $\rightsquigarrow$ $\nabla_0(G) = \mathrm{mad}(G)$.

For a **class** of graphs $\mathcal{C}$, we write:

$$\nabla_d(\mathcal{C}) \coloneqq \sup_{G \in \mathcal{C}} \nabla_d(G) \qquad \text{and} \qquad \omega_d(\mathcal{C}) \coloneqq \sup_{G \in \mathcal{C}} \omega_d(G).$$

Definition

$\mathcal{C}$ has **bounded expansion** if $\nabla_d(\mathcal{C})$ is finite for all $d \in \mathbb{N}$.

$\mathcal{C}$ is **nowhere dense** if $\omega_d(\mathcal{C})$ is finite for all $d \in \mathbb{N}$.

Notions of sparsity

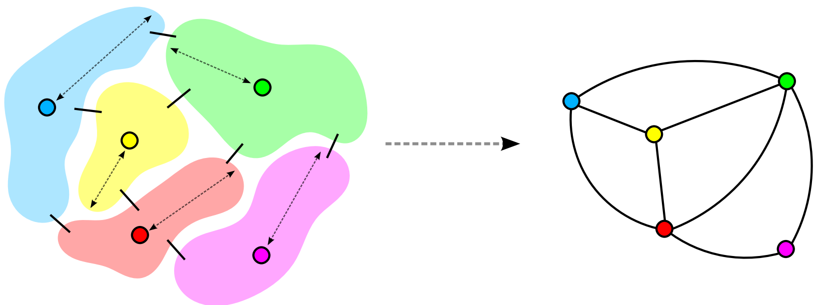
Equivalently:

Notions of sparsity

Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.

avgdeg$(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.



Notions of sparsity

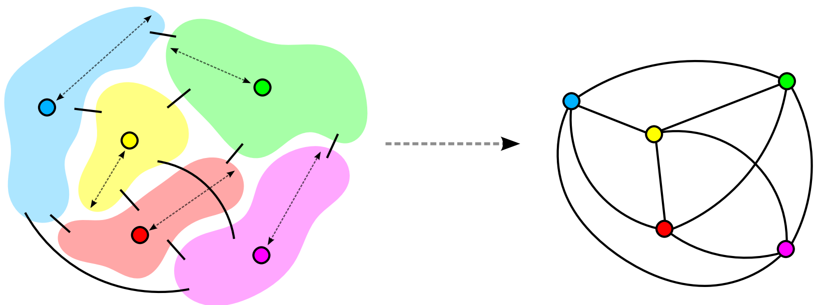
Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.

avgdeg$(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.

$\mathcal{C}$ is **nowhere dense** if for every $d \in \mathbb{N}$ there is $t(d) \in \mathbb{N}$ s.t.

$K_{t(d)}$ is not a depth-d minor of any $G \in \mathcal{C}$.



Notions of sparsity

Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.
$\text{avgdeg}(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.

$\mathcal{C}$ is **nowhere dense** if for every $d \in \mathbb{N}$ there is $t(d) \in \mathbb{N}$ s.t.
$K_{t(d)}$ is not a depth-d minor of any $G \in \mathcal{C}$.

Key idea: Sparsity is a property of a **class** of graphs.

Notions of sparsity

Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.

avgdeg$(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.

$\mathcal{C}$ is **nowhere dense** if for every $d \in \mathbb{N}$ there is $t(d) \in \mathbb{N}$ s.t.

$K_{t(d)}$ is not a depth-d minor of any $G \in \mathcal{C}$.

Key idea: Sparsity is a property of a **class** of graphs.

– It is a **limit property** of graphs from the class.

Notions of sparsity

Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.
$\mathrm{avgdeg}(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.

$\mathcal{C}$ is **nowhere dense** if for every $d \in \mathbb{N}$ there is $t(d) \in \mathbb{N}$ s.t.
$K_{t(d)}$ is not a depth-d minor of any $G \in \mathcal{C}$.

Key idea: **Sparsity** is a property of a **class** of graphs.

– It is a **limit property** of graphs from the class.

– Can be formalized using standard limit constructions (P, Toruńczyk).

Notions of sparsity

Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.
$\mathrm{avgdeg}(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.

$\mathcal{C}$ is **nowhere dense** if for every $d \in \mathbb{N}$ there is $t(d) \in \mathbb{N}$ s.t.
$K_{t(d)}$ is not a depth-d minor of any $G \in \mathcal{C}$.

Key idea: Sparsity is a property of a **class** of graphs.

- It is a **limit property** of graphs from the class.
- Can be formalized using standard limit constructions (P, Toruńczyk).

Every class with bnd degree, or excluding a minor, has bounded expansion.

Notions of sparsity

Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.

avgdeg$(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.

$\mathcal{C}$ is **nowhere dense** if for every $d \in \mathbb{N}$ there is $t(d) \in \mathbb{N}$ s.t.

$K_{t(d)}$ is not a depth-d minor of any $G \in \mathcal{C}$.

Key idea: Sparsity is a property of a **class** of graphs.

– It is a **limit property** of graphs from the class.

– Can be formalized using standard limit constructions (P, Toruńczyk).

Every class with bnd degree, or excluding a minor, has bounded expansion.

Every class of bounded expansion is nowhere dense, but **not** vice versa.

Notions of sparsity

Equivalently:

$\mathcal{C}$ has **bounded expansion** if for every $d \in \mathbb{N}$ there is $c(d) \in \mathbb{N}$ s.t.
$\mathrm{avgdeg}(H) \leqslant c(d)$ whenever H is a depth-d minor of some $G \in \mathcal{C}$.

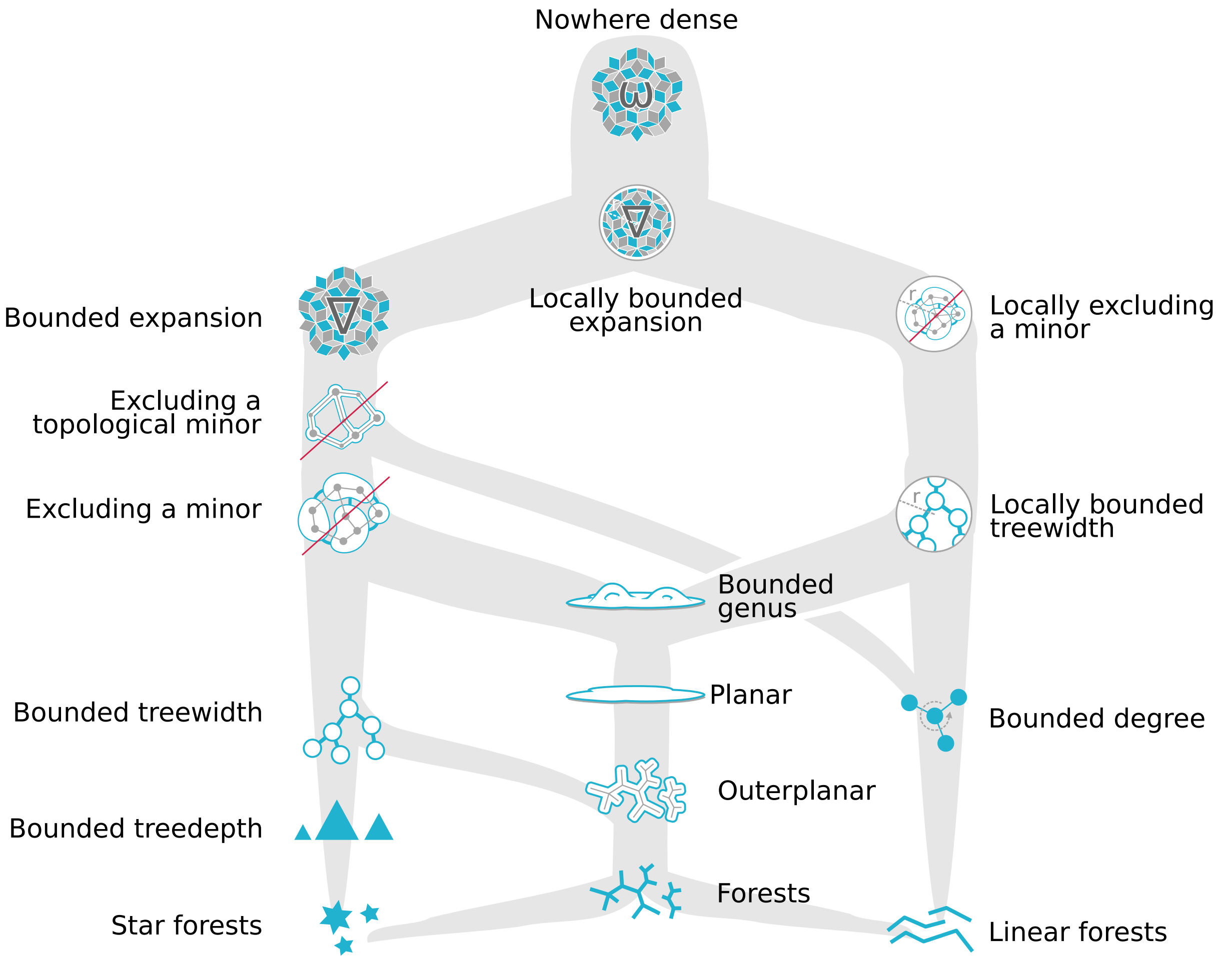
$\mathcal{C}$ is **nowhere dense** if for every $d \in \mathbb{N}$ there is $t(d) \in \mathbb{N}$ s.t.
$K_{t(d)}$ is not a depth-d minor of any $G \in \mathcal{C}$.

Key idea: Sparsity is a property of a **class** of graphs.

- It is a **limit property** of graphs from the class.
- Can be formalized using standard limit constructions (P, Toruńczyk).

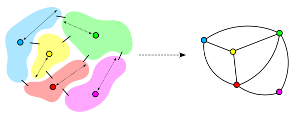
Every class with bnd degree, or excluding a minor, has bounded expansion.
Every class of bounded expansion is nowhere dense, but **not** vice versa.
Graphs from nowhere dense classes are somewhat sparse w.r.t. $\nabla_d(\cdot)$.

The World of Sparsity



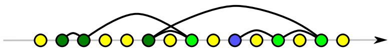
Equivalent characterizations

Sparsity of shallow minors

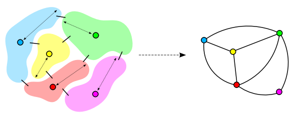


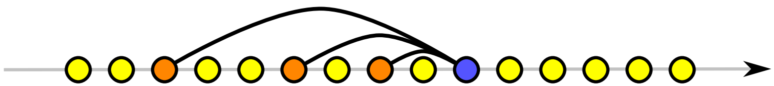
Equivalent characterizations

Generalized coloring numbers

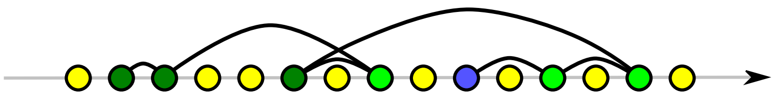


Sparsity of shallow minors





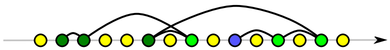
Degeneracy



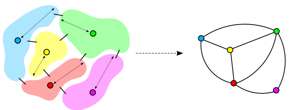
Weak coloring number

Equivalent characterizations

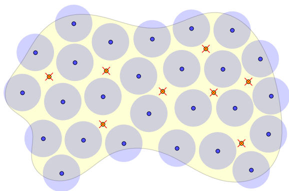
Generalized coloring numbers

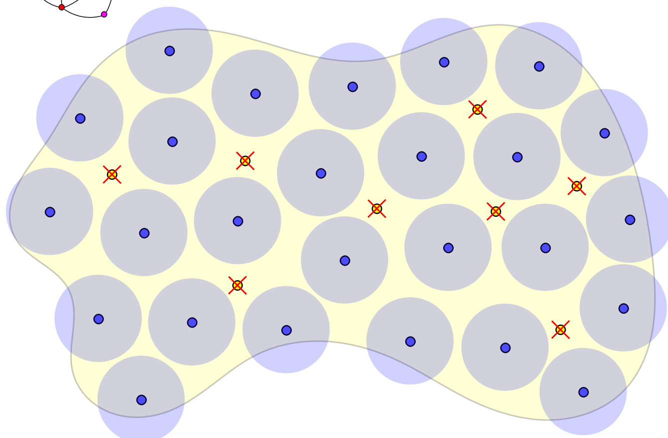


Sparsity of shallow minors



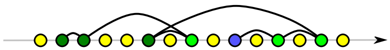
Uniform quasi-wideness

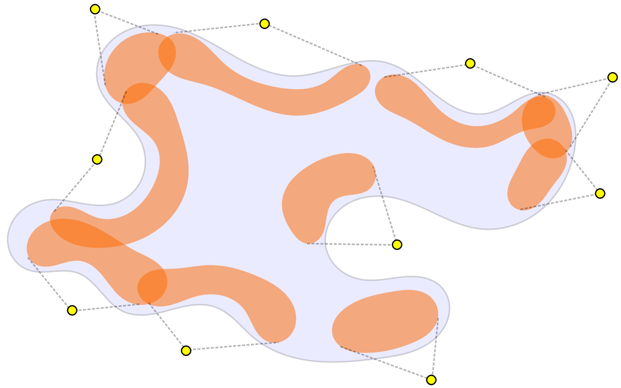




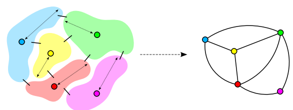
Equivalent characterizations

Generalized coloring numbers

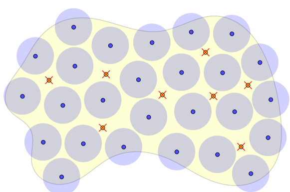




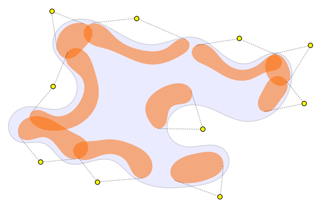
Sparsity of shallow minors



Uniform quasi-wideness



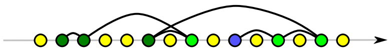
Neighborhood complexity



Equivalent characterizations

Stability

Generalized coloring numbers



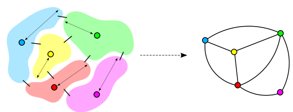
Sparsity of shallow top-minors

Fraternal augmentations

Neighborhood covers

Sparsity of shallow minors

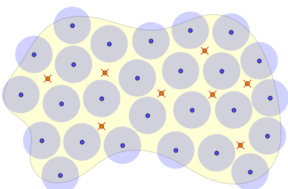
Low treedepth colorings



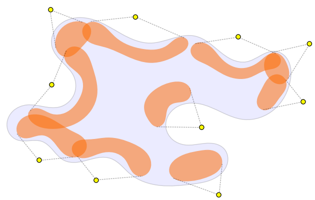
Uniform quasi-wideness

k-Helly property

Neighborhood complexity



Splitter game



Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

– Equivalence shows that we are working with fundamental notions.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

These properties can be used to design efficient **algorithms**.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

These properties can be used to design efficient **algorithms**.

- **Areas:** Parameterized, approximation, and distributed algorithms.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

These properties can be used to design efficient **algorithms**.

- **Areas:** Parameterized, approximation, and distributed algorithms.
- Applicable to problems of **local** nature.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

These properties can be used to design efficient **algorithms**.

- **Areas:** Parameterized, approximation, and distributed algorithms.
- Applicable to problems of **local** nature.

Sparsity delimits tractability of **First Order** **logic** on graphs.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

These properties can be used to design efficient **algorithms**.

- **Areas:** Parameterized, approximation, and distributed algorithms.
- Applicable to problems of **local** nature.

Sparsity delimits tractability of **First Order** **logic** on graphs.

- Provides connections with (algorithmic) finite model theory.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

These properties can be used to design efficient **algorithms**.

- **Areas:** Parameterized, approximation, and distributed algorithms.
- Applicable to problems of **local** nature.

Sparsity delimits tractability of **First Order** **logic** on graphs.

- Provides connections with (algorithmic) finite model theory.

Working with Sparsity

Many characterizations of **bnd expansion** and **nowhere denseness**.

- Equivalence shows that we are working with fundamental notions.
- Different characterizations present different perspectives on Sparsity.
- Each characterization is a **tool** applicable in different settings.

Original idea: Study the **combinatorics** of sparse graphs.

- **Goal:** Describe structural properties implied by sparsity.

These properties can be used to design efficient **algorithms**.

- **Areas:** Parameterized, approximation, and distributed algorithms.
- Applicable to problems of **local** nature.

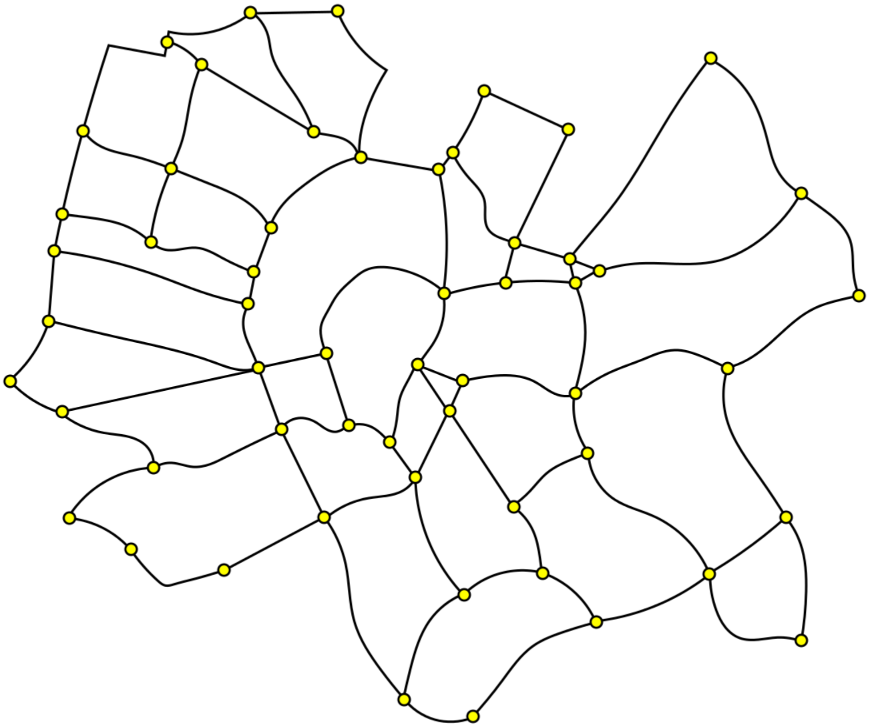
Sparsity delimits tractability of **First Order logic** on graphs.

- Provides connections with (algorithmic) finite model theory.

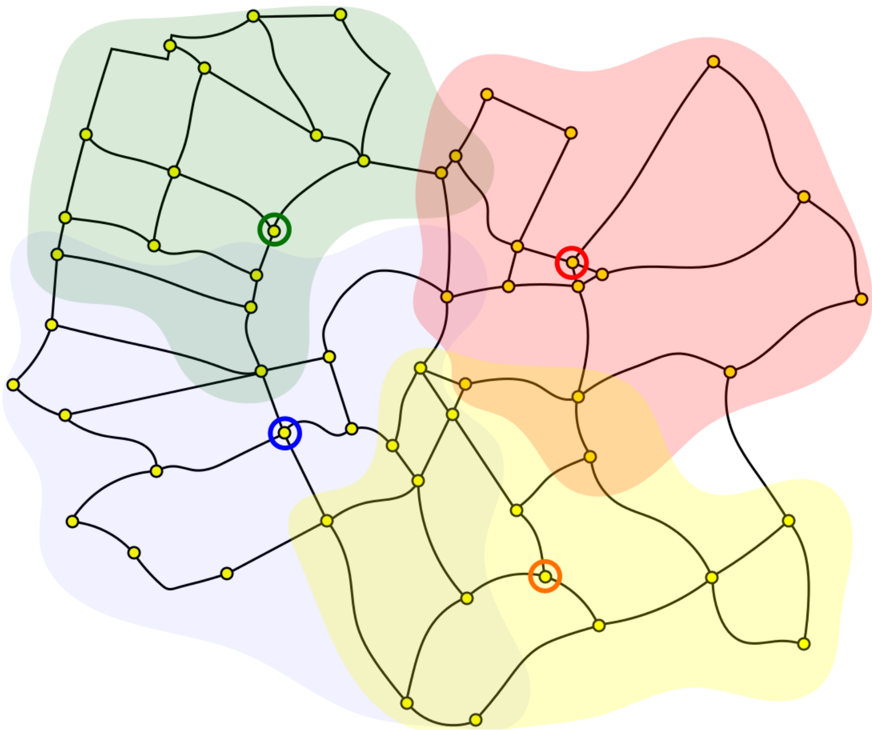
Now: Example **algorithmic** application.

Distance-d Dominating Set

Distance-*d* Dominating Set

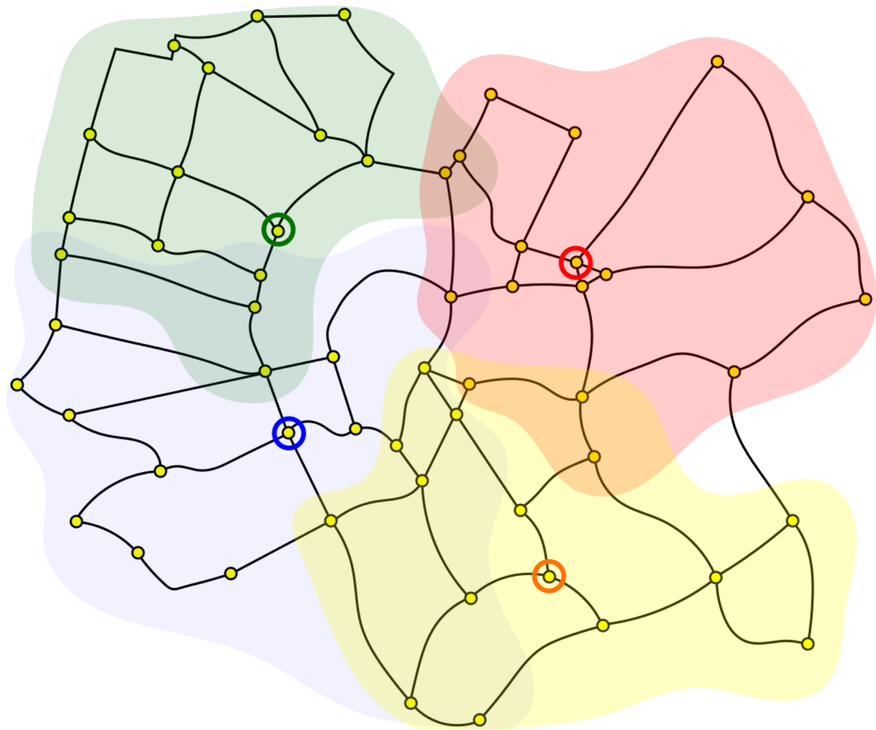


Distance-d Dominating Set



Distance-d dominating set:
set of vertices s.t. every vertex is at distance $\leqslant d$ from one of them

Distance-d Dominating Set



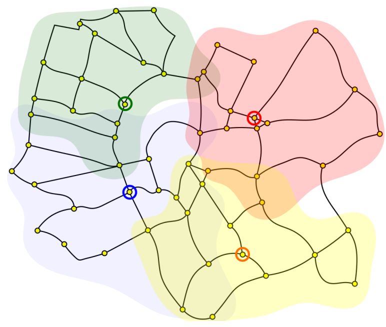
Distance-d dominating set:
set of vertices s.t. every vertex is at distance $\leqslant d$ from one of them

Here: A distance-3 dominating set of size 4.

Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

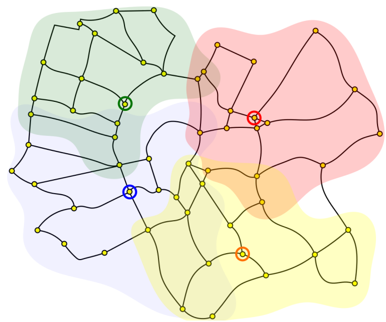


Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?



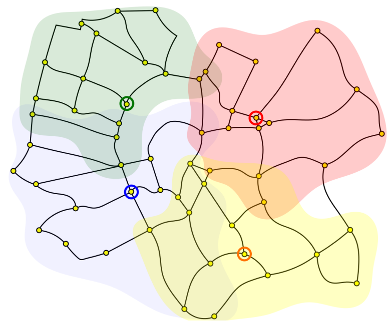
Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?

Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?



Distance-d Dominating Set

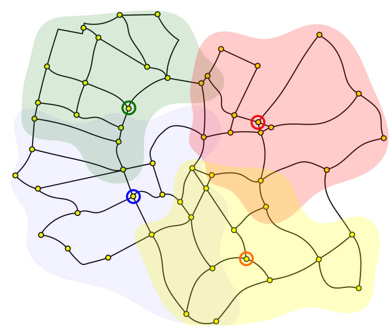
Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?

Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?

The problem is **NP-hard**, so probably no polytime algorithm.



Distance-d Dominating Set

Problem (Distance-d Dominating Set)

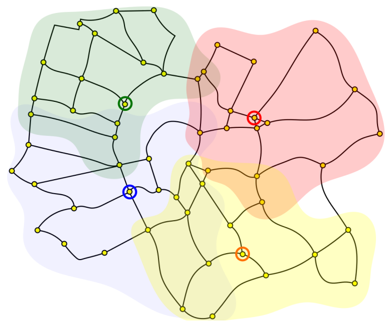
> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?

Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?

The problem is **NP-hard**, so probably no polytime algorithm.

Our setting: d is a fixed constant, k is small, G is sparse.



Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

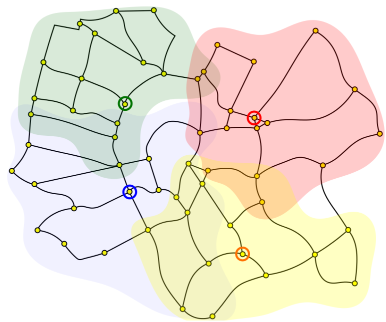
Ex 1. Can one place 4 police stations to cover the center of Cracow?

Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?

The problem is **NP-hard**, so probably no polytime algorithm.

Our setting: d is a fixed constant, k is small, G is sparse.

Naive: $\mathcal{O}(n^k)$ time, where $n = |V(G)|$.



Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?

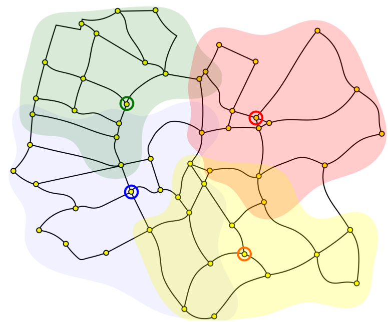
Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?

The problem is **NP-hard**, so probably no polytime algorithm.

Our setting: d is a fixed constant, k is small, G is sparse.

Naive: $\mathcal{O}(n^k)$ time, where $n = |V(G)|$.

– For arbitrary G, no $\mathcal{O}(n^{k-\varepsilon})$-time algorithm for any $\varepsilon > 0$.



Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?

Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?

The problem is **NP-hard**, so probably no polytime algorithm.

Our setting: d is a fixed constant, k is small, G is sparse.

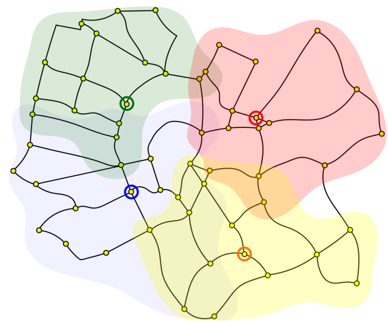
Naive: $\mathcal{O}(n^k)$ time, where $n = |V(G)|$.

– For arbitrary G, no $\mathcal{O}(n^{k-\varepsilon})$-time algorithm for any $\varepsilon > 0$.

Now: For any nowhere dense class $\mathcal{C}$,

a $k^{\mathcal{O}(k)} \cdot \|G\|$-time algorithm for $G \in \mathcal{C}$.

$\|G\| = |V(G)| + |E(G)|$



Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?

Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?

The problem is **NP-hard**, so probably no polytime algorithm.

Our setting: d is a fixed constant, k is small, G is sparse.

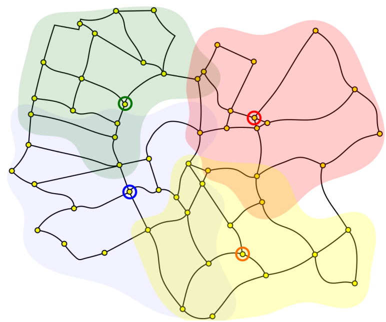
Naive: $\mathcal{O}(n^k)$ time, where $n = |V(G)|$.

– For arbitrary G, no $\mathcal{O}(n^{k-\varepsilon})$-time algorithm for any $\varepsilon > 0$.

Now: For any nowhere dense class $\mathcal{C}$,

a $k^{\mathcal{O}(k)} \cdot \|G\|$-time algorithm for $G \in \mathcal{C}$.

$\|G\| = |V(G)| + |E(G)|$



– Distinction between **XP** and **FPT** running times.

Distance-d Dominating Set

Problem (Distance-d Dominating Set)

> Given G and $k \in \mathbb{N}$, does G admit a distance-d dominating set of size k?

Ex 1. Can one place 4 police stations to cover the center of Cracow?

Ex 2. Can one divide Poland into 16 voivodeships so that every place is at distance 100km from the capital of its voivodeship?

The problem is **NP-hard**, so probably no polytime algorithm.

Our setting: d is a fixed constant, k is small, G is sparse.

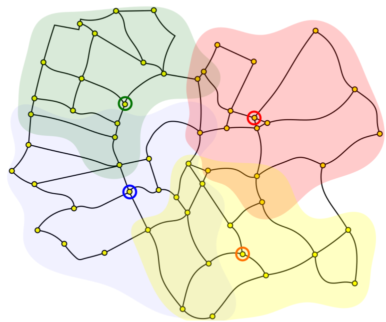
Naive: $\mathcal{O}(n^k)$ time, where $n = |V(G)|$.

– For arbitrary G, no $\mathcal{O}(n^{k-\varepsilon})$-time algorithm for any $\varepsilon > 0$.

Now: For any nowhere dense class $\mathcal{C}$,

a $k^{\mathcal{O}(k)} \cdot \|G\|$-time algorithm for $G \in \mathcal{C}$.

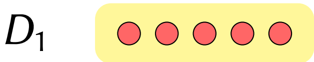
$\|G\| = |V(G)| + |E(G)|$



– Distinction between **XP** and **FPT** running times.

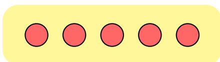
The Algorithm

The Algorithm



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

The Algorithm

D_1 

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

If D_1 dominates the whole graph, terminate and return **YES**.

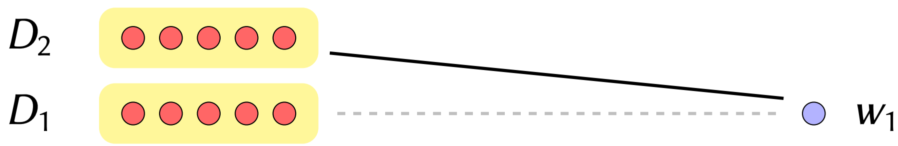
The Algorithm

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

If D_1 dominates the whole graph, terminate and return **YES**.

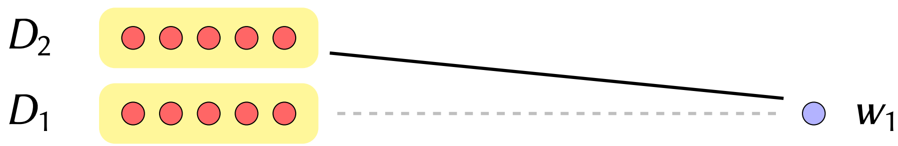
Otherwise, some w_1 is not dominated.

The Algorithm



Round 2: Take any k-tuple of vertices D_2 that dominates w_1.

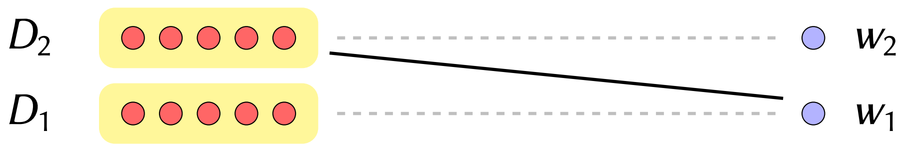
The Algorithm



Round 2: Take any k-tuple of vertices D_2 that dominates w_1.

If D_2 dominates the whole graph, terminate and return **YES**.

The Algorithm

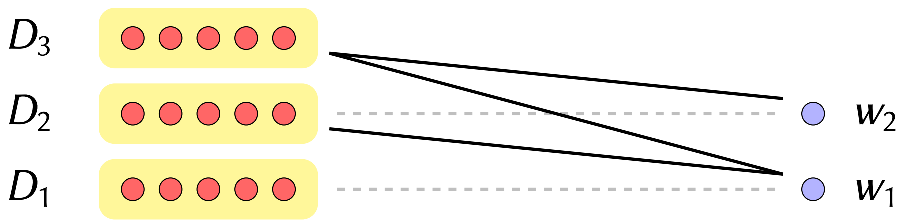


Round 2: Take any k-tuple of vertices D_2 that dominates w_1.

If D_2 dominates the whole graph, terminate and return **YES**.

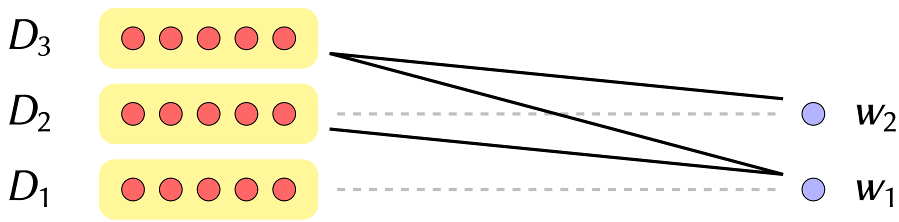
Otherwise, some w_2 is not dominated.

The Algorithm



Round 3: Take any k-tuple of vertices D_3 that dominates $\{w_1, w_2\}$.

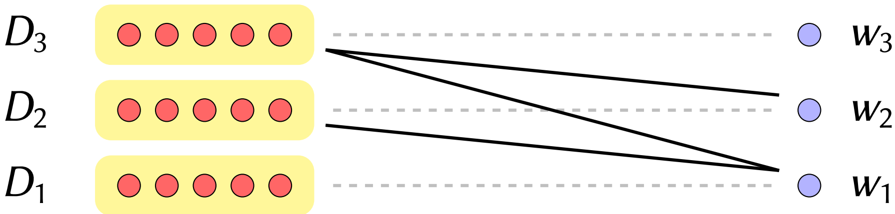
The Algorithm



Round 3: Take any k-tuple of vertices D_3 that dominates $\{w_1, w_2\}$.

If D_3 dominates the whole graph, terminate and return **YES**.

The Algorithm

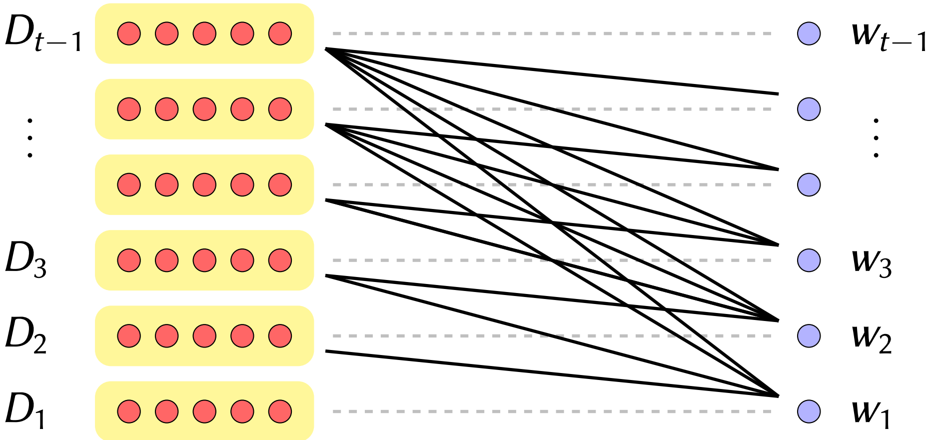


Round 3: Take any k-tuple of vertices D_3 that dominates $\{w_1, w_2\}$.

If D_3 dominates the whole graph, terminate and return **YES**.

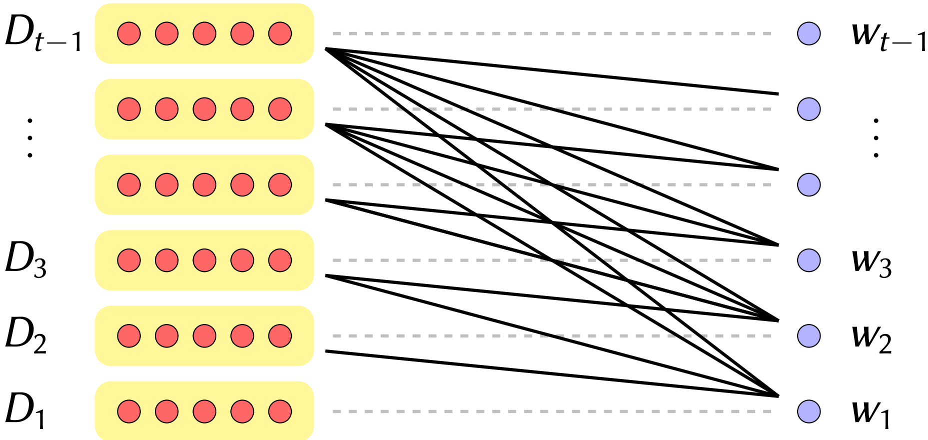
Otherwise, some w_3 is not dominated.

The Algorithm



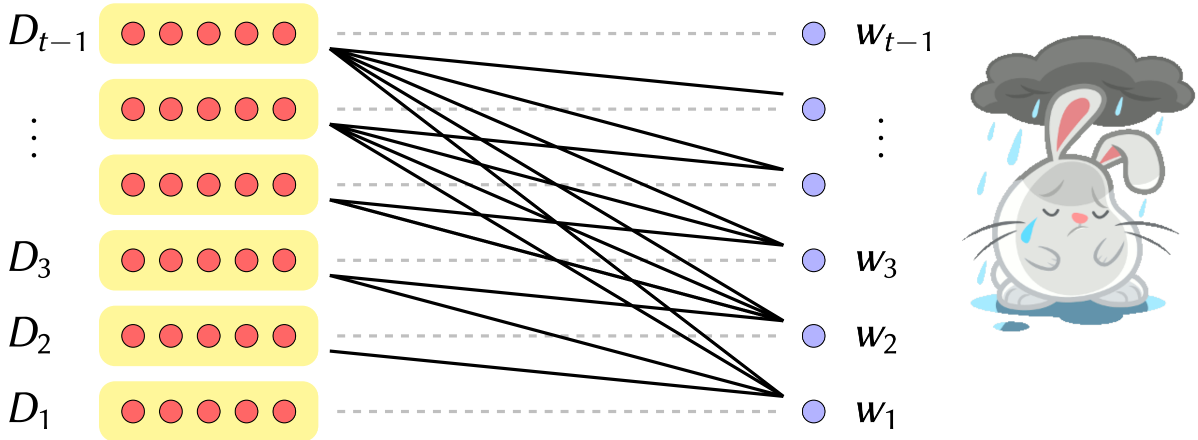
Round t:

The Algorithm



Round t: Try to find any k-tuple D_t that dominates $\{w_1, \ldots, w_{t-1}\}$.

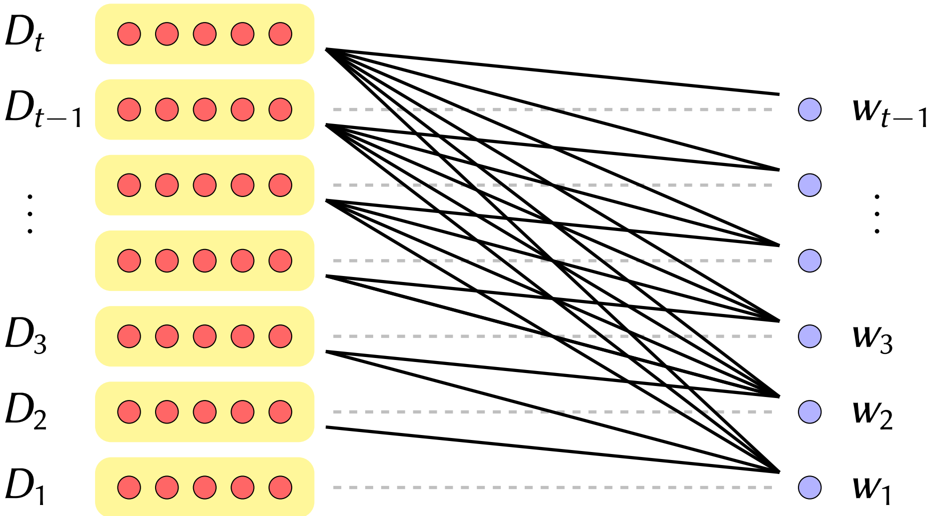
The Algorithm



Round t: Try to find any k-tuple D_t that dominates $\{w_1, \ldots, w_{t-1}\}$.

If there is none, terminate and return **NO**.

The Algorithm

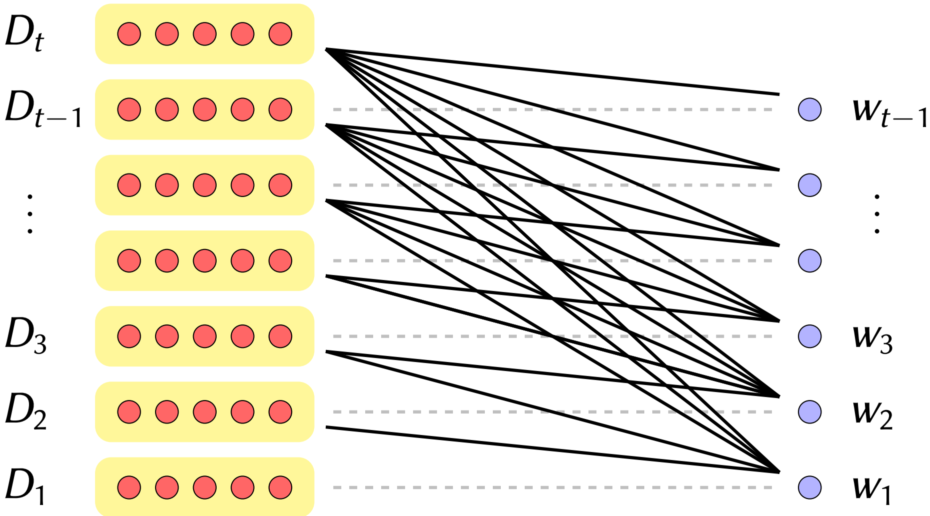


Round t: Try to find any k-tuple D_t that dominates $\{w_1, \ldots, w_{t-1}\}$.

> If there is none, terminate and return **NO**.

Otherwise, take any such k-tuple D_t.

The Algorithm



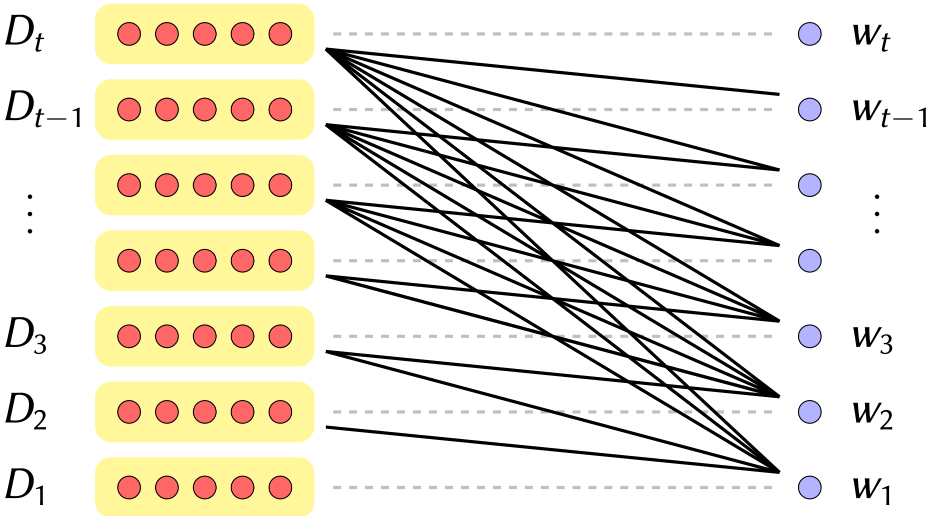
Round t: Try to find any k-tuple D_t that dominates $\{w_1, \ldots, w_{t-1}\}$.

> If there is none, terminate and return **NO**.

Otherwise, take any such k-tuple D_t.

> If D_t dominates G, terminate and return **YES**.

The Algorithm



Round t: Try to find any k-tuple D_t that dominates $\{w_1, \ldots, w_{t-1}\}$.

> If there is none, terminate and return **NO**.

Otherwise, take any such k-tuple D_t.

> If D_t dominates G, terminate and return **YES**.

Otherwise, some w_t is not dominated, and **Proceed**.

Analysis

Analysis

Problem in round t:

> Find k vertices that distance-d dominate $\{w_1, \ldots, w_{t-1}\}$.

Analysis

Problem in round t:

> Find k vertices that distance-d dominate $\{w_1, \ldots, w_{t-1}\}$.

This problem can be solved in time $t^{\mathcal{O}(k)} \cdot \|G\|$.

Analysis

Problem in round t:

> Find k vertices that distance-d dominate $\{w_1, \ldots, w_{t-1}\}$.

This problem can be solved in time $t^{\mathcal{O}(k)} \cdot \|G\|$.

Ergo: Running time $L^{\mathcal{O}(k)} \cdot \|G\|$, where L is the total number of rounds.

Analysis

Problem in round t:

> Find k vertices that distance-d dominate $\{w_1, \ldots, w_{t-1}\}$.

This problem can be solved in time $t^{\mathcal{O}(k)} \cdot \|G\|$.

Ergo: Running time $L^{\mathcal{O}(k)} \cdot \|G\|$, where L is the total number of rounds.

Question: Why the number of rounds should be bounded?

Analysis

Problem in round t:

> Find k vertices that distance-d dominate $\{w_1, \ldots, w_{t-1}\}$.

This problem can be solved in time $t^{\mathcal{O}(k)} \cdot \|G\|$.

Ergo: Running time $L^{\mathcal{O}(k)} \cdot \|G\|$, where L is the total number of rounds.

Question: Why the number of rounds should be bounded?

Theorem (Fabiański, P, Siebertz, Toruńczyk)

For every nowhere dense class $\mathcal{C}$ and $d \in \mathbb{N}$, there is a polynomial $p(\cdot)$ such that the algorithm run on any $G \in \mathcal{C}$ and k, performs $\leqslant p(k)$ rounds.

Analysis

Problem in round t:

> Find k vertices that distance-d dominate $\{w_1, \ldots, w_{t-1}\}$.

This problem can be solved in time $t^{\mathcal{O}(k)} \cdot \|G\|$.

Ergo: Running time $L^{\mathcal{O}(k)} \cdot \|G\|$, where L is the total number of rounds.

Question: Why the number of rounds should be bounded?

Theorem (Fabiański, P, Siebertz, Toruńczyk)

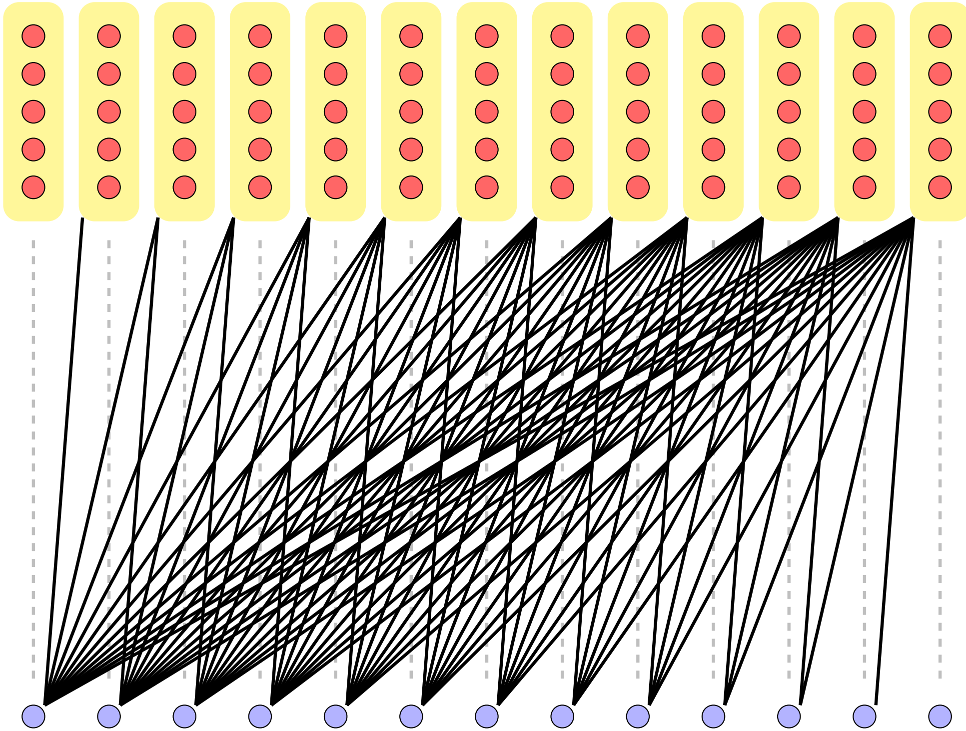
For every nowhere dense class $\mathcal{C}$ and $d \in \mathbb{N}$, there is a polynomial $p(\cdot)$ such that the algorithm run on any $G \in \mathcal{C}$ and k, performs $\leqslant p(k)$ rounds.

Cor: Runtime $p(k)^{\mathcal{O}(k)} \cdot \|G\| = k^{\mathcal{O}(k)} \cdot \|G\|$ on any nowhere dense $\mathcal{C}$.

Proof sketch

Proof sketch

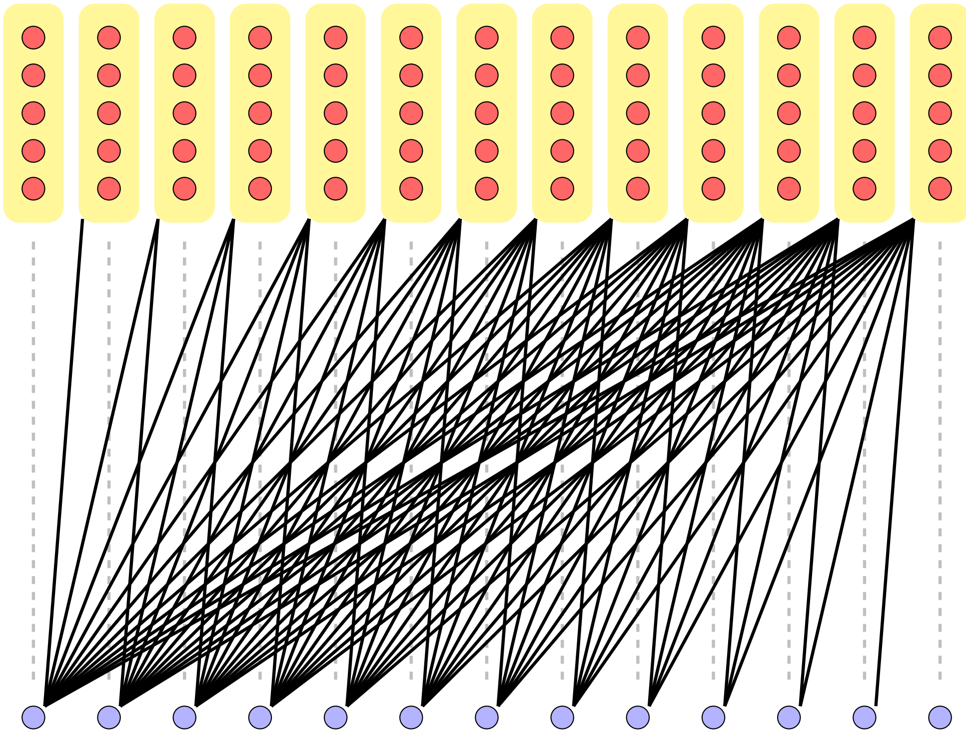
The Algorithm constructs a **semi-ladder** of length L.



Proof sketch

The Algorithm constructs a **semi-ladder** of length L.

– Every D_i dominates all w_j with $j < i$, but not w_i.

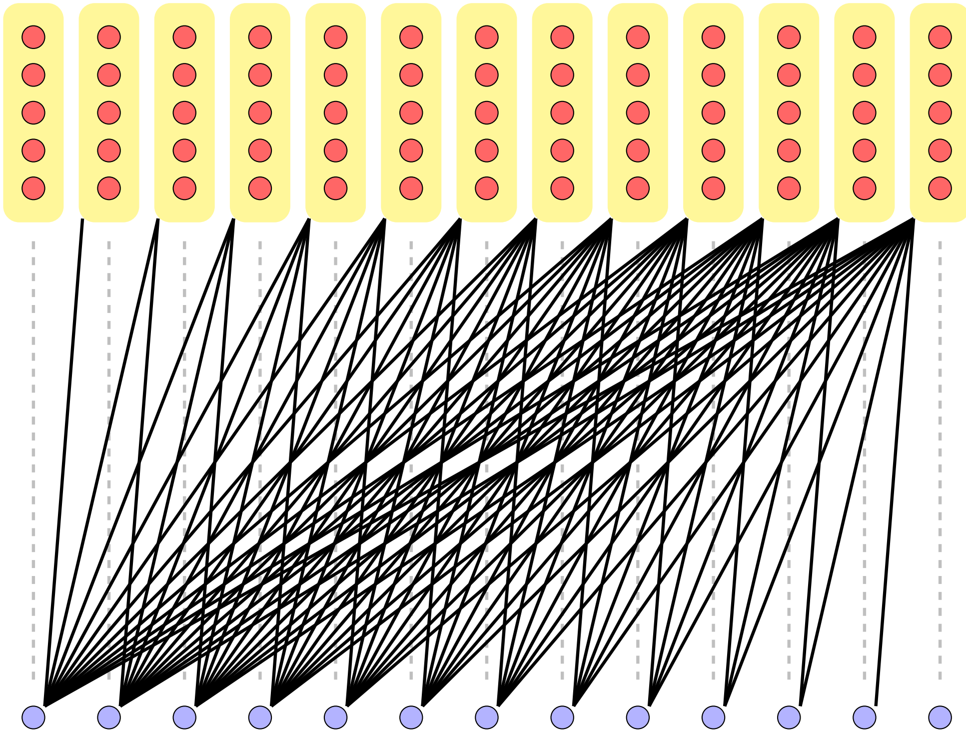


Proof sketch

The Algorithm constructs a **semi-ladder** of length L.

– Every D_i dominates all w_j with $j < i$, but not w_i.

Uniform quasi-wideness $\Rightarrow$ Such a structure cannot be too long.



Generalizing problems

Distance-d Dominating Set is a problem with **local** constraints.

Generalizing problems

Distance-d Dominating Set is a problem with **local** constraints.

Q: What other problems have this **locality** property?

Generalizing problems

Distance-*d* Dominating Set is a problem with **local** constraints.

Q: What other problems have this **locality** property?

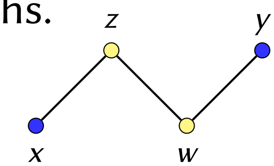
A: Problems expressible in **First Order Logic** on graphs.

Generalizing problems

Distance-d Dominating Set is a problem with **local** constraints.

Q: What other problems have this **locality** property?

A: Problems expressible in **First Order Logic** on graphs.



Ex 1. $\delta_d(x, y)$ expresses that $\mathrm{dist}(x, y) \leqslant d$.

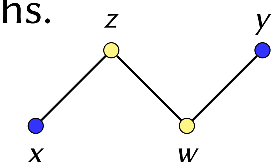
$$\delta_3(x, y) \quad := \quad \delta_2(x, y) \vee \exists_z \exists_w \, \mathrm{adj}(x, z) \wedge \mathrm{adj}(z, w) \wedge \mathrm{adj}(w, y).$$

Generalizing problems

Distance-d Dominating Set is a problem with **local** constraints.

Q: What other problems have this **locality** property?

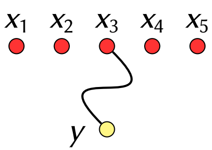
A: Problems expressible in **First Order Logic** on graphs.



Ex 1. $\delta_d(x, y)$ expresses that $\mathrm{dist}(x, y) \leqslant d$.

$$\delta_3(x, y) \quad := \quad \delta_2(x, y) \vee \exists_z \exists_w \, \mathrm{adj}(x, z) \wedge \mathrm{adj}(z, w) \wedge \mathrm{adj}(w, y).$$

Ex 2. $\varphi_{d,k}$ expresses that there is a distance-d dominating set of size k.



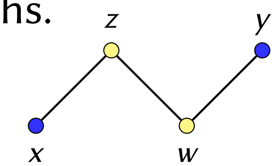
$$\varphi_{d,k} \quad := \quad \exists_{x_1} \exists_{x_2} \ldots \exists_{x_k} \forall_y \bigvee_{i=1}^{k} \delta_d(y, x_i).$$

Generalizing problems

Distance-d Dominating Set is a problem with **local** constraints.

Q: What other problems have this **locality** property?

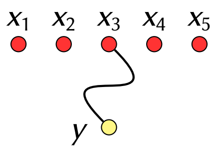
A: Problems expressible in **First Order Logic** on graphs.



Ex 1. $\delta_d(x, y)$ expresses that $\mathrm{dist}(x, y) \leqslant d$.

$$\delta_3(x, y) \quad := \quad \delta_2(x, y) \vee \exists_z \exists_w \,\mathrm{adj}(x, z) \wedge \mathrm{adj}(z, w) \wedge \mathrm{adj}(w, y).$$

Ex 2. $\varphi_{d,k}$ expresses that there is a distance-d dominating set of size k.



$$\varphi_{d,k} \quad := \quad \exists_{x_1} \exists_{x_2} \ldots \exists_{x_k} \forall_y \bigvee_{i=1}^{k} \delta_d(y, x_i).$$

Theorem (Gaifman)

First Order Logic on graphs is **local**:

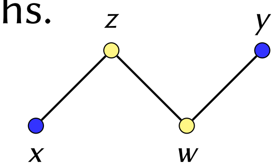
It cannot express much more than the above.

Generalizing problems

Distance-d Dominating Set is a problem with **local** constraints.

Q: What other problems have this **locality** property?

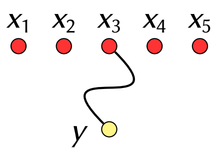
A: Problems expressible in **First Order Logic** on graphs.



Ex 1. $\delta_d(x, y)$ expresses that $\mathrm{dist}(x, y) \leqslant d$.

$$\delta_3(x, y) \quad := \quad \delta_2(x, y) \vee \exists_z \exists_w \,\mathrm{adj}(x, z) \wedge \mathrm{adj}(z, w) \wedge \mathrm{adj}(w, y).$$

Ex 2. $\varphi_{d,k}$ expresses that there is a distance-d dominating set of size k.



$$\varphi_{d,k} \quad := \quad \exists_{x_1} \exists_{x_2} \ldots \exists_{x_k} \forall_y \bigvee_{i=1}^{k} \delta_d(y, x_i).$$

Theorem (Gaifman)

First Order Logic on graphs is **local**:

It cannot express much more than the above.

Q: Can we solve efficiently all FO-expressible problems on sparse graphs?

FO model checking

Problem (FO model checking)

> Given a graph G and an FO sentence φ, does φ hold in G?

FO model checking

Problem (FO model checking)

> Given a graph G and an FO sentence φ, does φ hold in G?

Naive: Running time $\mathcal{O}(\|G\|^{\|\varphi\|})$.

FO model checking

Problem (FO model checking)

> Given a graph G and an FO sentence φ, does φ hold in G?

Naive: Running time $\mathcal{O}(\|G\|^{\|\varphi\|})$.

- Degree of the polynomial has to depend on $\|\varphi\|$, unless $\mathsf{FPT} = \mathsf{AW}[\star]$.

FO model checking

Problem (FO model checking)

> Given a graph G and an FO sentence φ, does φ hold in G?

Naive: Running time $\mathcal{O}(\|G\|^{\|\varphi\|})$.

– Degree of the polynomial has to depend on $\|\varphi\|$, unless $\mathrm{FPT} = \mathrm{AW}[\star]$.

Q: Can we do better in sparse graphs?

FO model checking

Problem (FO model checking)

> Given a graph G and an FO sentence φ, does φ hold in G?

Naive: Running time $\mathcal{O}(\|G\|^{\|\varphi\|})$.

– Degree of the polynomial has to depend on $\|\varphi\|$, unless $\mathsf{FPT} = \mathsf{AW}[\star]$.

Q: Can we do better in sparse graphs?

Theorem (Grohe, Kreutzer, Siebertz)

For every nowhere dense class $\mathcal{C}$, FO model checking on graphs from $\mathcal{C}$ can be done in time $f(\varphi) \cdot \|G\|^{1+\varepsilon}$, for some function f and any $\varepsilon > 0$.

FO model checking

Problem (FO model checking)

> Given a graph G and an FO sentence φ, does φ hold in G?

Naive: Running time $\mathcal{O}(\|G\|^{\|\varphi\|})$.

– Degree of the polynomial has to depend on $\|\varphi\|$, unless FPT $=$ AW[$\star$].

Q: Can we do better in sparse graphs?

Theorem (Grohe, Kreutzer, Siebertz)

For every nowhere dense class $\mathcal{C}$, FO model checking on graphs from $\mathcal{C}$ can be done in time $f(\varphi) \cdot \|G\|^{1+\varepsilon}$, for some function f and any $\varepsilon > 0$.

Theorem (Dvořák, Král', Thomas)

If $\mathcal{C}$ is **not** nowhere dense, and is closed under taking subgraphs, then FO model checking on $\mathcal{C}$ is as hard as on general graphs.

FO model checking

Problem (FO model checking)

> Given a graph G and an FO sentence φ, does φ hold in G?

Naive: Running time $\mathcal{O}(\|G\|^{\|\varphi\|})$.

– Degree of the polynomial has to depend on $\|\varphi\|$, unless FPT $=$ AW$[\star]$.

Q: Can we do better in sparse graphs?

Theorem (Grohe, Kreutzer, Siebertz)

For every nowhere dense class $\mathcal{C}$, FO model checking on graphs from $\mathcal{C}$ can be done in time $f(\varphi) \cdot \|G\|^{1+\varepsilon}$, for some function f and any $\varepsilon > 0$.

Theorem (Dvořák, Král', Thomas)

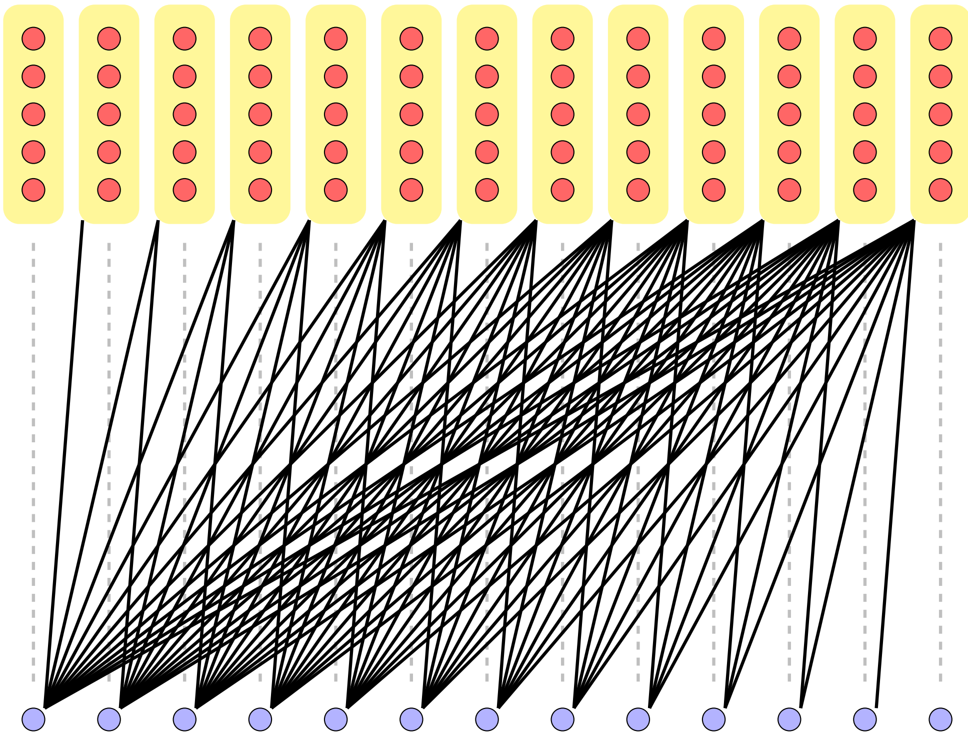
If $\mathcal{C}$ is **not** nowhere dense, and is closed under taking subgraphs, then FO model checking on $\mathcal{C}$ is as hard as on general graphs.

Cor: Nowhere denseness **exactly** delimits tractability of **First Order Logic**.

Stability

Stability

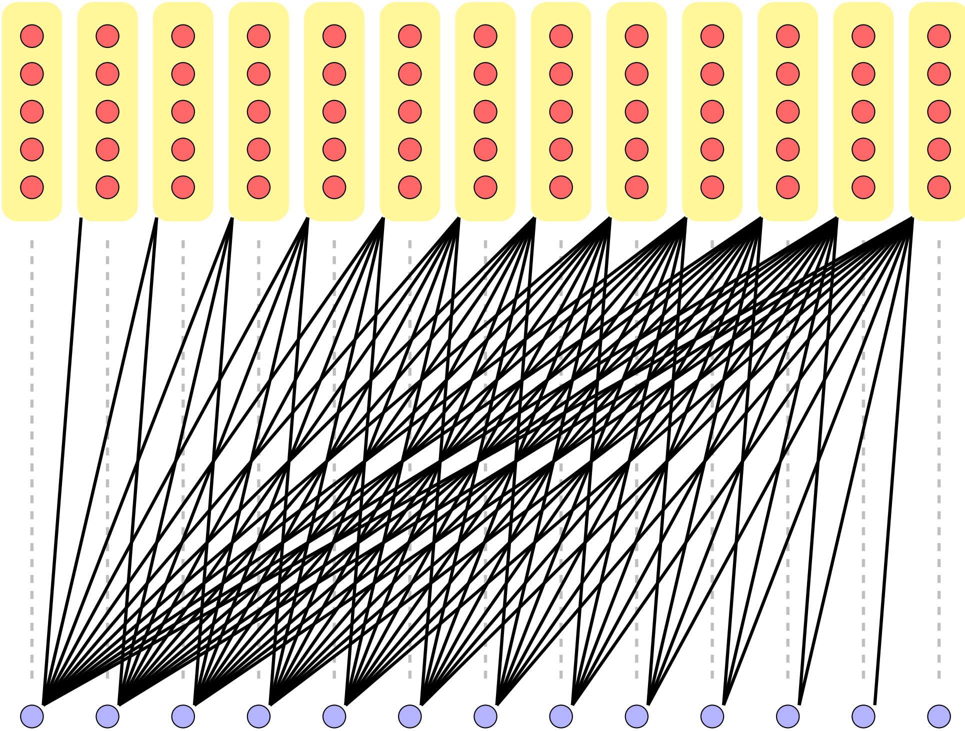
Idea: Generalize the bound on semi-ladders to other properties?



Stability

Idea: Generalize the bound on semi-ladders to other properties?

– Distance-d domination $\leftrightsquigarrow$ $\psi_{d,k}(\bar{x}, y) = \bar{x}$ distance-d dominates y.



Stability

Idea: Generalize the bound on semi-ladders to other properties?

– Distance-d domination $\leftrightsquigarrow$ $\psi_{d,k}(\bar{x}, y) = \bar{x}$ distance-d dominates y.

Definition

$\bar{x}, \bar{y}$: tuples of variables, $\varphi(\bar{x}, \bar{y})$: an FO formula

Stability

Idea: Generalize the bound on semi-ladders to other properties?

– Distance-d domination $\leftrightsquigarrow$ $\psi_{d,k}(\bar{x}, y) = \bar{x}$ distance-d dominates y.

Definition

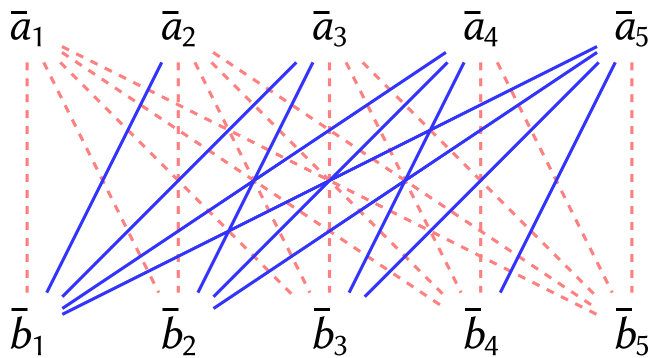
$\bar{x}, \bar{y}$: tuples of variables, $\varphi(\bar{x}, \bar{y})$: an FO formula

A **φ-ladder** of length ℓ in G consists of evaluations

$$\bar{a}_1, \ldots, \bar{a}_\ell \qquad \text{and} \qquad \bar{b}_1, \ldots, \bar{b}_\ell$$

of $\bar{x}$ and $\bar{y}$, respectively, such that

$$\varphi(\bar{a}_i, \bar{b}_j) \text{ holds} \qquad \Leftrightarrow \qquad i > j.$$



Stability

Idea: Generalize the bound on semi-ladders to other properties?

– Distance-d domination $\rightsquigarrow$ $\psi_{d,k}(\bar{x}, y) = \bar{x}$ distance-d dominates y.

Definition

$\bar{x}, \bar{y}$: tuples of variables, $\varphi(\bar{x}, \bar{y})$: an FO formula

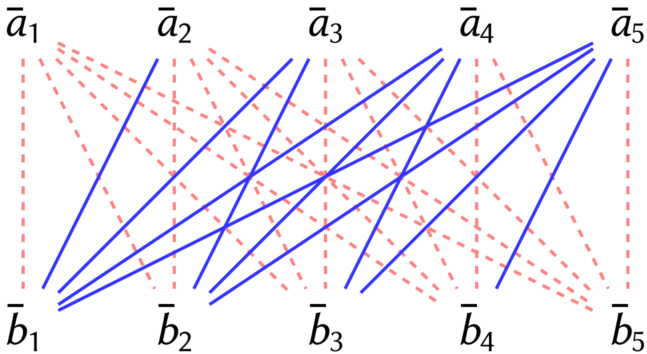
A **φ-ladder** of length ℓ in G consists of evaluations

$$\bar{a}_1, \ldots, \bar{a}_\ell \quad \text{and} \quad \bar{b}_1, \ldots, \bar{b}_\ell$$

of $\bar{x}$ and $\bar{y}$, respectively, such that

$$\varphi(\bar{a}_i, \bar{b}_j) \text{ holds} \quad \Leftrightarrow \quad i > j.$$

NB: More restrictive than semi-ladders!



Stability

Stability

Definition

Ladder index of φ on $\mathcal{C}$:=

supremum of lengths of φ-ladders in graphs from $\mathcal{C}$.

Stability

Definition

Ladder index of φ on $\mathcal{C}$:=

supremum of lengths of φ-ladders in graphs from $\mathcal{C}$.

Definition

A class $\mathcal{C}$ is **stable** if every FO formula has a finite ladder index on $\mathcal{C}$.

Stability

Definition

Ladder index of φ on $\mathcal{C}$:=

supremum of lengths of φ-ladders in graphs from $\mathcal{C}$.

Definition

A class $\mathcal{C}$ is **stable** if every FO formula has a finite ladder index on $\mathcal{C}$.

Intuition: In stable classes one cannot define arbitrary long **linear orders**.

Stability

Definition

Ladder index of φ on $\mathcal{C}$:=

supremum of lengths of φ-ladders in graphs from $\mathcal{C}$.

Definition

A class $\mathcal{C}$ is **stable** if every FO formula has a finite ladder index on $\mathcal{C}$.

Intuition: In stable classes one cannot define arbitrary long **linear orders**.

Theorem (Adler and Adler)

A subgraph-closed graph class $\mathcal{C}$ is nowhere dense if and only if it is stable.

Stability

Definition

Ladder index of φ on $\mathcal{C}$:=

supremum of lengths of φ-ladders in graphs from $\mathcal{C}$.

Definition

A class $\mathcal{C}$ is **stable** if every FO formula has a finite ladder index on $\mathcal{C}$.

Intuition: In stable classes one cannot define arbitrary long **linear orders**.

Theorem (Adler and Adler)

A subgraph-closed graph class $\mathcal{C}$ is nowhere dense if and only if it is stable.

Note: There are plenty of stable classes that are not nowhere dense.

Stability

Definition

Ladder index of φ on $\mathcal{C}$:=

supremum of lengths of φ-ladders in graphs from $\mathcal{C}$.

Definition

A class $\mathcal{C}$ is **stable** if every FO formula has a finite ladder index on $\mathcal{C}$.

Intuition: In stable classes one cannot define arbitrary long **linear orders**.

Theorem (Adler and Adler)

A subgraph-closed graph class $\mathcal{C}$ is nowhere dense if and only if it is stable.

Note: There are plenty of stable classes that are not nowhere dense.

– They consist of well-structured, dense graphs.

Stability and Sparsity

Stability and Sparsity

Stability is an established area of **model theory**.

Stability and Sparsity

Stability is an established area of **model theory**.

– **Focus:** dividing lines between **simple** and **complex** theories.

Stability and Sparsity

Stability is an established area of **model theory**.

- **Focus:** dividing lines between **simple** and **complex** theories.
- Many concepts and techniques have **analogues** in **Sparsity**.

Stability and Sparsity

Stability is an established area of **model theory**.

- **Focus:** dividing lines between **simple** and **complex** theories.
- Many concepts and techniques have **analogues** in **Sparsity**.
- **Reason:** **Stability** projects to **Sparsity** under subgraph-closeness.

Stability and Sparsity

Stability is an established area of **model theory**.

- **Focus:** dividing lines between **simple** and **complex** theories.
- Many concepts and techniques have **analogues** in **Sparsity**.
- **Reason:** **Stability** projects to **Sparsity** under subgraph-closeness.
- Enables a **transfer** of ideas and techniques.

Stability and Sparsity

Stability is an established area of **model theory**.

- **Focus:** dividing lines between **simple** and **complex** theories.
- Many concepts and techniques have **analogues** in **Sparsity**.
- **Reason:** **Stability** projects to **Sparsity** under subgraph-closeness.
- Enables a **transfer** of ideas and techniques.

Current direction: A theory for well-structured dense graphs.

Stability and Sparsity

Stability is an established area of **model theory**.

- **Focus:** dividing lines between **simple** and **complex** theories.
- Many concepts and techniques have **analogues** in **Sparsity**.
- **Reason:** **Stability** projects to **Sparsity** under subgraph-closeness.
- Enables a **transfer** of ideas and techniques.

Current direction: A theory for well-structured dense graphs.

- **Objective:** Structural understanding of stable classes of graphs.

Stability and Sparsity

Stability is an established area of **model theory**.

- **Focus:** dividing lines between **simple** and **complex** theories.
- Many concepts and techniques have **analogues** in **Sparsity**.
- **Reason:** **Stability** projects to **Sparsity** under subgraph-closeness.
- Enables a **transfer** of ideas and techniques.

Current direction: A theory for well-structured dense graphs.

- **Objective:** Structural understanding of stable classes of graphs.
- **Algorithmic** applications in arbitrary stable classes?

Sparsity in Warsaw

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → **Bremen**

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → **Bremen**

Graduated students: Grzegorz Fabiański, Wojciech Nadara, Adam Paszke

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → **Bremen**

Graduated students: Grzegorz Fabiański, Wojciech Nadara, Adam Paszke

Sparsity course offered for master students in CS

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → **Bremen**

Graduated students: Grzegorz Fabiański, Wojciech Nadara, Adam Paszke

Sparsity course offered for master students in CS

– Three editions: Winter semesters 2012/13, 2017/18, and 2019/20

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → Bremen

Graduated students: Grzegorz Fabiański, Wojciech Nadara, Adam Paszke

Sparsity course offered for master students in CS

– Three editions: Winter semesters 2012/13, 2017/18, and 2019/20

– **Webpages:** `www.mimuw.edu.pl/~mp248287/sparsity` and `/sparsity2`

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → Bremen

Graduated students: Grzegorz Fabiański, Wojciech Nadara, Adam Paszke

Sparsity course offered for master students in CS

- Three editions: Winter semesters 2012/13, 2017/18, and 2019/20
- **Webpages:** `www.mimuw.edu.pl/~mp248287/sparsity` and `/sparsity2`
- Extensive **lecture notes** on the topic

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → Bremen

Graduated students: Grzegorz Fabiański, Wojciech Nadara, Adam Paszke

Sparsity course offered for master students in CS

– Three editions: Winter semesters 2012/13, 2017/18, and 2019/20

– **Webpages:** `www.mimuw.edu.pl/~mp248287/sparsity` and `/sparsity2`

– Extensive **lecture notes** on the topic

– **New:** Video recordings of lectures on Youtube.

Sparsity in Warsaw

Marcin Pilipczuk

Sebastian Siebertz

Szymon Toruńczyk

Berlin → **Bremen**

Graduated students: Grzegorz Fabiański, Wojciech Nadara, Adam Paszke

Sparsity course offered for master students in CS

- Three editions: Winter semesters 2012/13, 2017/18, and 2019/20
- **Webpages:** `www.mimuw.edu.pl/~mp248287/sparsity` and `/sparsity2`
- Extensive **lecture notes** on the topic
- **New:** Video recordings of lectures on Youtube.

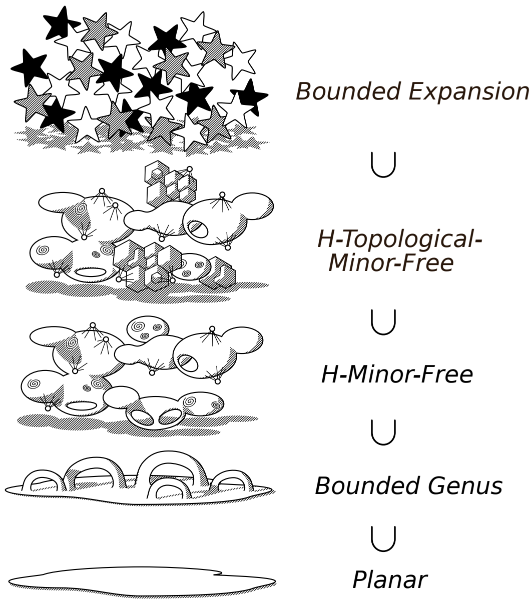
In two weeks, a one-week **crash course** as part of **ALGOMANET**.

A big **Thank You** to:

all my Sparsity coauthors, especially Marcin, Sebi, and Szymon

Felix Reidl, for all the pretty figures in this talk

Michał Skrzypczak, for BeamerikZ



A big **Thank You** to:

all my Sparsity coauthors, especially Marcin, Sebi, and Szymon

Felix Reidl, for all the pretty figures in this talk

Michał Skrzypczak, for `Beamer`*ikZ*

Thank you for your attention!