A Logic-Based Algorithmic Meta-Theorem for Mim-Width

Benjamin Bergougnoux.
Joint work with **Jan Dreier**, **Lars Jaffke**

Séminaire de l'équipe Optimisation Combinatoire, Grenoble
November 24, 2022

Parameterized Complexity

A problem is **hard**! What to do?

- **Approximation**

Parameterized Complexity

A problem is **hard**! What to do?

- **Approximation**
- Look at **restrictions** of this problem

Parameterized Complexity

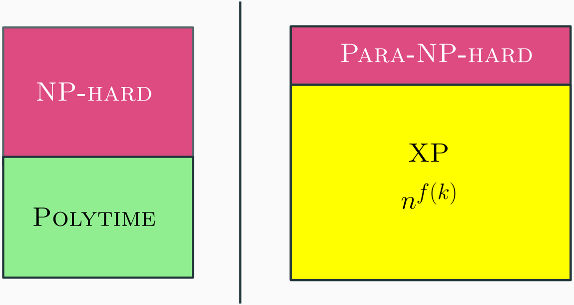
A problem is **hard**! What to do?

- **Approximation**
- Look at **restrictions** of this problem
  - Classical approach: inputs satisfying some properties

Parameterized Complexity

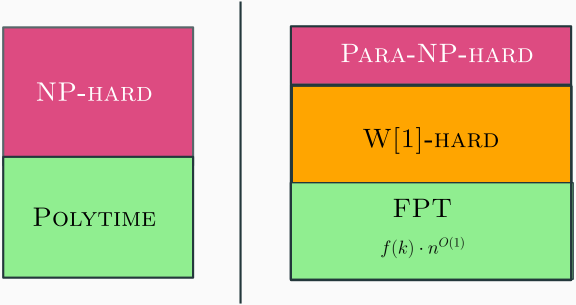
A problem is **hard**! What to do?

- **Approximation**
- Look at **restrictions** of this problem
  - Classical approach: inputs satisfying some properties
  - **Parameterized approach**



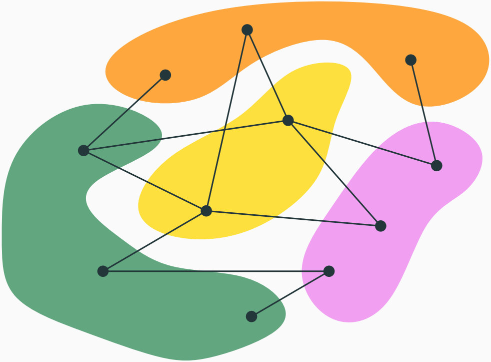
A problem is **hard**! What to do?

- **Approximation**
- Look at **restrictions** of this problem
  - Classical approach: inputs satisfying some properties
  - **Parameterized approach**



Para-NP-hard

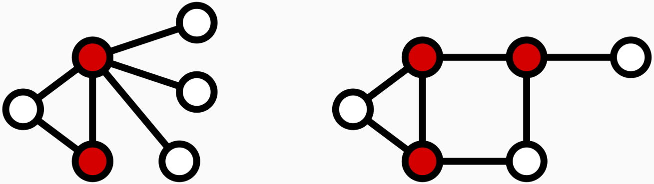
Graph Coloring parameterized by k the **number of colors**



NP-hard for $k = 3$.

FPT

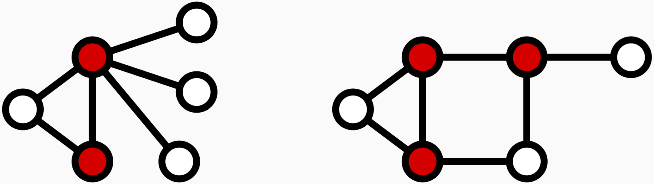
Vertex Cover parameterized by k the **solution size**



Can be solved in time $2^{O(k)}n$.

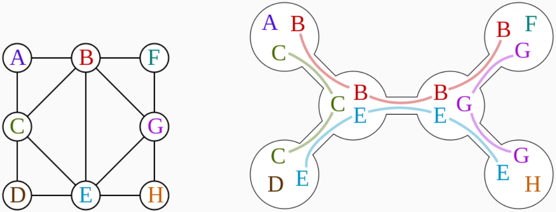
XP and W[1]-hard

Independent Set parameterized by k the **solution size**



- Can be solved in time $O(n^{k+1})$
- **Unlikely** to be **FPT** (unless W[1]=FPT)

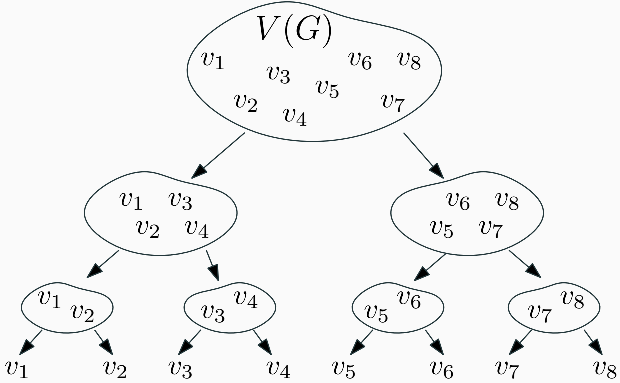
Width parameters



Tree-width, clique-width, rank-width, mim-width...

- Measure the **structural complexity** of graphs
- Give **efficient** algorithms for many NP-**hard** problems

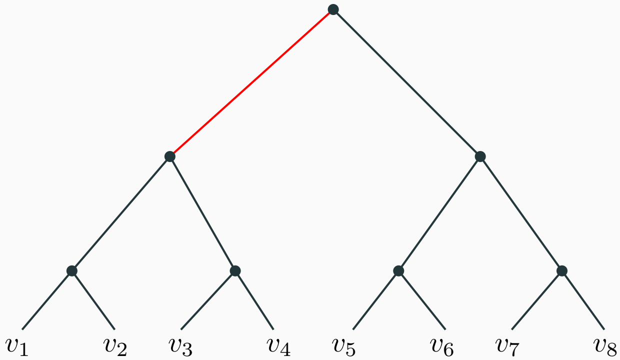
Recursively decompose a graph into **simple** cuts



Branch-decomposition: recursively **cut the vertex set** in **two**

Graph decomposition

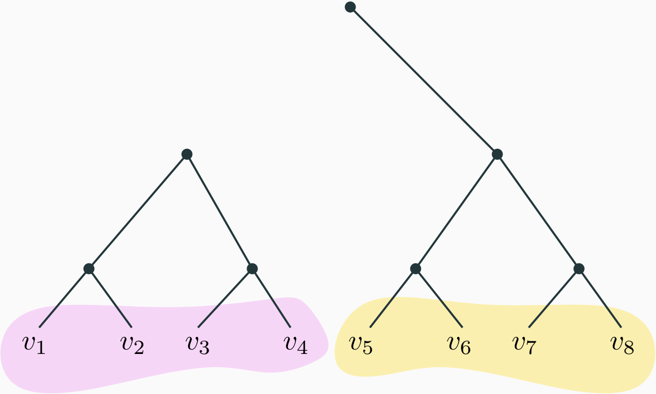
Recursively decompose a graph into **simple** cuts



Branch-decomposition: recursively **cut the vertex set** in **two**

Graph decomposition

Recursively decompose a graph into **simple** cuts

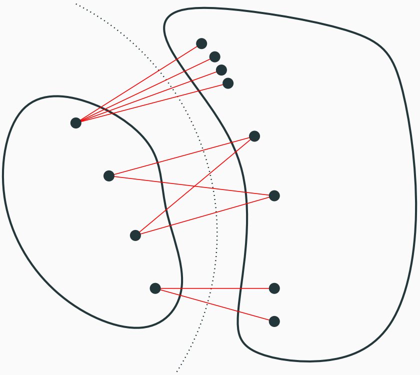


Branch-decomposition: recursively **cut the vertex set** in **two**

Graph decomposition

Recursively decompose a graph into **simple** cuts

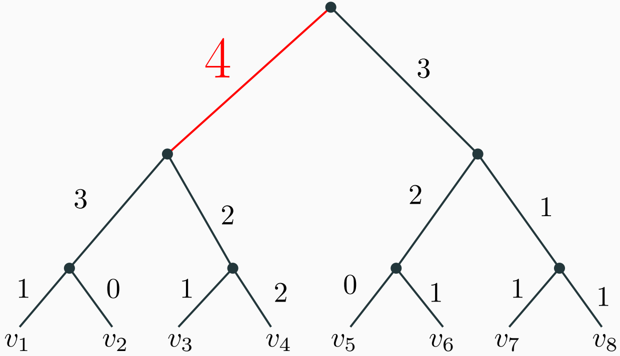
Simplicity of cuts is measured with a **cut function** f: **cut** $\to \mathbb{N}$.



Different notions of **simplicity** $=$ different **width parameters**.

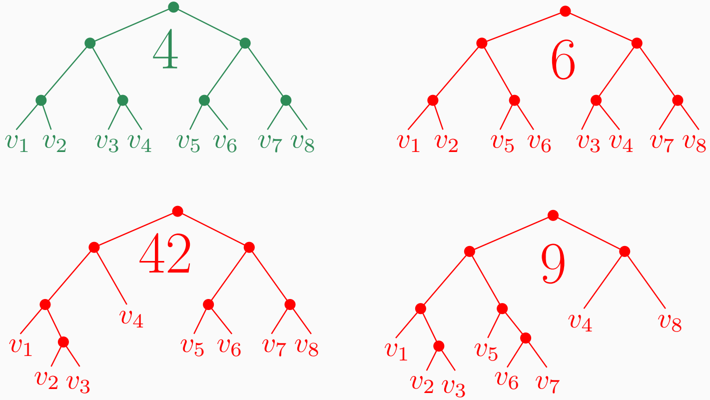
Graph decomposition

Width of a **decomposition** $D =$ **max** f(cut) over the **cuts** of D.



Graph decomposition

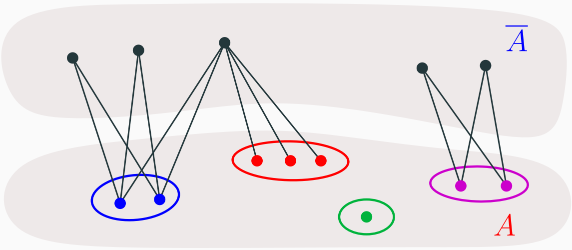
Width of a **graph** $G =$ **min** **widths** of its **decompositions**.



Width of a **graph class** $\mathcal{C} =$ **max** **widths** of $G \in \mathcal{C}$.

Module-width [Rao, 2006]

Defined from the function mw$(A) := |\{N(v) \cap \overline{A} \mid v \in A\}|$.



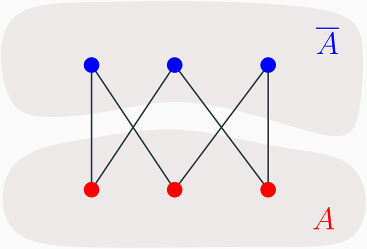
Linearly equivalent to clique-width:

Theorem [Rao, 2006]

For all graphs G, we have mw$(G) \leqslant$ cw$(G) \leqslant 2$mw(G).

Rank-width [Oum, 2005]

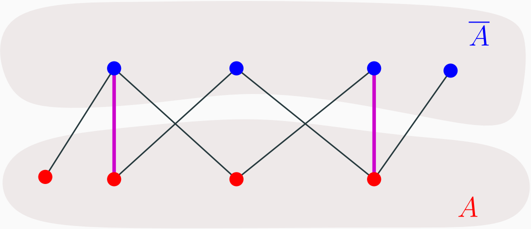
Defined from the function rw(A) := the rank of adjacency matrix between A and $\overline{A}$ over the **binary field**.



$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Mim-width [Vatshelle, 2012]

Defined from the function mim(A) := the size of a maximum induced matching in the bipartite graph between A and $\overline{A}$.



Main property

For every $X \subseteq A$, there exits $X' \subseteq A$ such that $|X'| \leqslant$ mim(A) and $N(X) \cap \overline{A} = N(X') \cap \overline{A}$.

Comparing these widths

- **Modeling power**
- **$\Rightarrow$Algorithmic applications$\Leftarrow$**
- **Complexity of computing a good decomposition**
  - NP-hard!

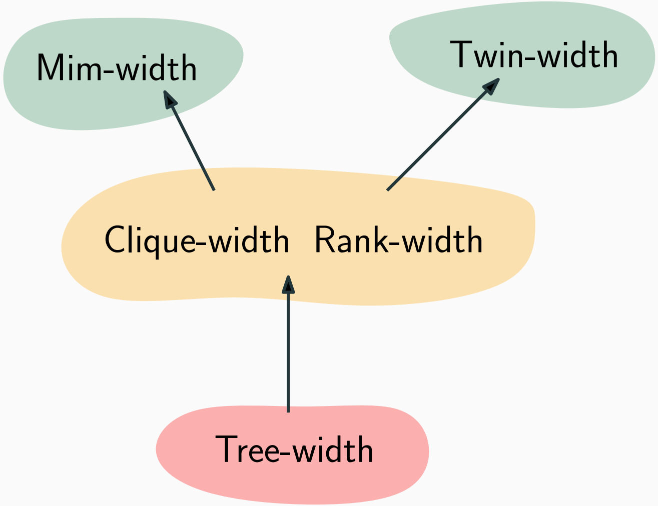
Comparing these widths

- **Modeling power**
- **$\Rightarrow$Algorithmic applications$\Leftarrow$**
- **Complexity of computing a good decomposition**
  - NP-hard!
  - We know **efficient FPT** approximation algorithms for **tree-width** and **rank-width** with runtime $2^{O(k)} \cdot n^{O(1)}$.

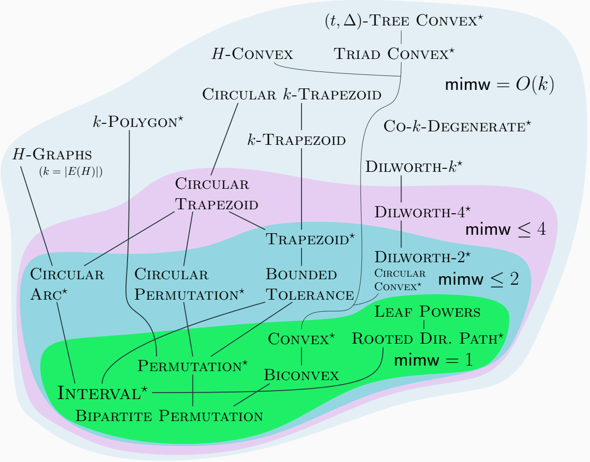
Comparing these widths

- **Modeling power**
- $\Rightarrow$**Algorithmic applications**$\Leftarrow$
- **Complexity of computing a good decomposition**
  - NP-hard!
  - We know **efficient FPT** approximation algorithms for **tree-width** and **rank-width** with runtime $2^{O(k)} \cdot n^{O(1)}$.
  - Tough open questions for **mim-width**!

Modeling Power



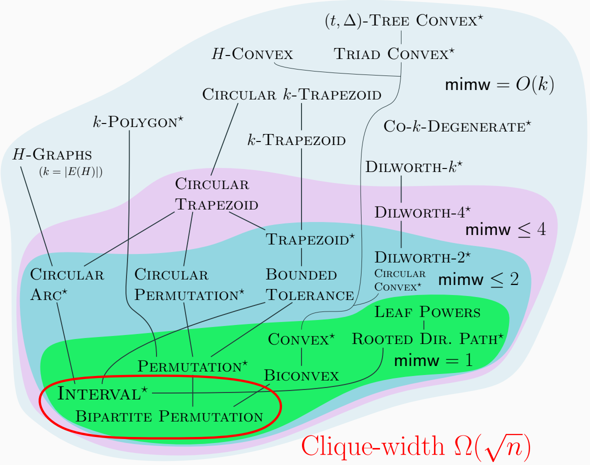
Modeling Power



Plus several H-free and (H_1, H_2)-free graph classes

[Brettel et al., 2022] [Munaro and Yang, 2022]

Modeling Power



Plus several H-free and (H_1, H_2)-free graph classes

[Brettel et al., 2022] [Munaro and Yang, 2022]

Algorithmic applications of mim-width

Locally Checkable Vertex Subset (LCVS)

Independent Set	Induced d-Regular Subgraph
Dominating Set	Perfect Code
Induced Matching	Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring	Odd Cycle Transversal
H-Homomorphism	Perfect Matching Cut
H-Covering	. . .

[Bui-Xuan, Telle and Vatshelle, 2013]

Algorithmic applications of mim-width

Locally Checkable Vertex Subset (LCVS)

Independent Set
Induced d-Regular Subgraph
Dominating Set
Perfect Code
Induced Matching
Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring
Odd Cycle Transversal
H-Homomorphism
Perfect Matching Cut
H-Covering
• • •

[Bui-Xuan, Telle and Vatshelle, 2013]

Induced Path Problems

Induced Disjoint Paths
Longest Induced Path
H-Induced Topological Minor

[Jaffke, Kwon, and Telle, 2017]

[Jaffke, Kwon, and Telle, 2018]

Feedback Vertex Set

Algorithmic applications of mim-width

Locally Checkable Vertex Subset (LCVS)

INDEPENDENT SET
INDUCED d-REGULAR SUBGRAPH
DOMINATING SET
PERFECT CODE
INDUCED MATCHING
TOTAL DOMINATING SET

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-COLORING
ODD CYCLE TRANSVERSAL
H-HOMOMORPHISM
PERFECT MATCHING CUT
H-COVERING
• • •

[Bui-Xuan, Telle and Vatshelle, 2013]

Induced Path Problems

INDUCED DISJOINT PATHS
LONGEST INDUCED PATH
H-INDUCED TOPOLOGICAL MINOR

[Jaffke, Kwon, and Telle, 2017]

[Jaffke, Kwon, and Telle, 2018]

FEEDBACK VERTEX SET

CONNECTED, ACYCLIC LCVS

CONNECTED, ACYCLIC LCVP

[Bergougnoux and Kante, 2019]

Algorithmic applications of mim-width

<u>Locally Checkable Vertex Subset (LCVS)</u>

Independent Set — Induced d-Regular Subgraph
Dominating Set — Perfect Code
Induced Matching — Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

<u>Locally Checkable Vertex Partitioning (LCVP)</u>

k-Coloring — Odd Cycle Transversal
H-Homomorphism — Perfect Matching Cut
H-Covering — ...

[Bui-Xuan, Telle and Vatshelle, 2013]

[Jaffke, Kwon, and Telle, 2018]

Feedback Vertex Set

<u>Connected, Acyclic LCVS</u>

<u>Connected, Acyclic LCVP</u>

[Bergougnoux and Kante, 2019]

<u>Induced Path Problems</u>

Induced Disjoint Paths — Longest Induced Path
H-Induced Topological Minor

[Jaffke, Kwon, and Telle, 2017]

Distance-r LCVS
Distance-r LCVP

[Jaffke et al. 2018]

Algorithmic applications of mim-width

Locally Checkable Vertex Subset (LCVS)

Independent Set — Induced d-Regular Subgraph
Dominating Set — Perfect Code
Induced Matching — Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring — Odd Cycle Transversal
H-Homomorphism — Perfect Matching Cut
H-Covering — • • •

[Bui-Xuan, Telle and Vatshelle, 2013]

Induced Path Problems

Induced Disjoint Paths — Longest Induced Path
H-Induced Topological Minor

[Jaffke, Kwon, and Telle, 2017]

[Jaffke, Kwon, and Telle, 2018]

Feedback Vertex Set

Connected, Acyclic LCVS

Connected, Acyclic LCVP

[Bergougnoux and Kante, 2019]

Distance-r LCVS
Distance-r LCVP

[Jaffke et al. 2018]

Semitotal Dominating Set
[Galby, Munaro and Ries, 2020]

Algorithmic applications of mim-width

Locally Checkable Vertex Subset (LCVS)

Independent Set — Induced d-Regular Subgraph
Dominating Set — Perfect Code
Induced Matching — Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring — Odd Cycle Transversal
H-Homomorphism — Perfect Matching Cut
H-Covering — ...

[Bui-Xuan, Telle and Vatshelle, 2013]

Induced Path Problems

Induced Disjoint Paths — Longest Induced Path
H-Induced Topological Minor

[Jaffke, Kwon, and Telle, 2017]

[Jaffke, Kwon, and Telle, 2018]

Feedback Vertex Set

Connected, Acyclic LCVS

Connected, Acyclic LCVP

[Bergougnoux and Kante, 2019]

Node Multiway Cut — Subset Feedback Vertex Set

[Bergougnoux, Papadopoulos and Telle, 2020]

Distance-r LCVS
Distance-r LCVP

[Jaffke et al. 2018]

Semitotal Dominating Set

[Galby, Munaro and Ries, 2020]

Algorithmic applications of mim-width

Locally Checkable Vertex Subset (LCVS)

Independent Set, Induced d-Regular Subgraph, Dominating Set, Perfect Code, Induced Matching, Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring, Odd Cycle Transversal, H-Homomorphism, Perfect Matching Cut, H-Covering, . . .

[Bui-Xuan, Telle and Vatshelle, 2013]

[Jaffke, Kwon, and Telle, 2018]

Feedback Vertex Set

Induced Path Problems

Induced Disjoint Paths, Longest Induced Path

H-Induced Topological Minor

[Jaffke, Kwon, and Telle, 2017]

Connected, Acyclic LCVS

Connected, Acyclic LCVP

[Bergougnoux and Kante, 2019]

Node Multiway Cut, Subset Feedback Vertex Set

[Bergougnoux, Papadopoulos and Telle, 2020]

Distance-r LCVS

Distance-r LCVP

[Jaffke et al. 2018]

Semitotal Dominating Set

[Galby, Munaro and Ries, 2020]

q-b-Coloring, $[k]$-Roman Domination, Conflict-Free q-Coloring

d-stable Locally Checkable problems

[Gonzales and Mann, 2022]

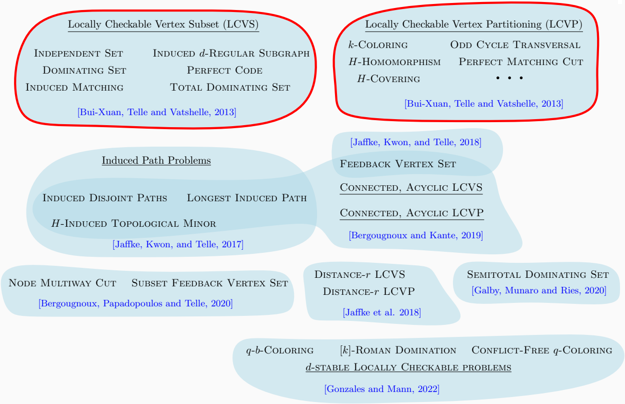
Width parameters and meta-theorems

mim-width ???	twin-width	FO
clique-width	rank-width	MSO_1
treewidth	branchwidth	MSO_2

Algorithmic Meta-Theorems

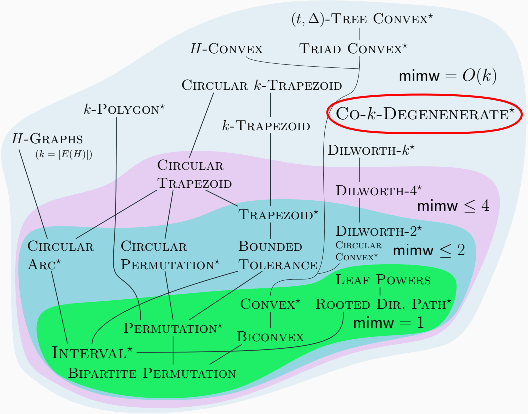
Each problem expressible in logic L is **efficiently** solvable on graphs of bounded $*$-width (given a decomposition).

Requirements for the target logic



- Should capture **Independent Set**

Requirements for the target logic



- Should capture **Independent Set**
- But not **Clique**: **para-NP-hard** given a decomposition

Existential MSO$_1$

- Vertex variables (x, y, z)

Existential MSO$_1$

- Vertex variables (x, y, z)
- Vertex-set variables (X, Y, Z)

Existential MSO_1

- Vertex variables (x, y, z)
- Vertex-set variables (X, Y, Z)
- Vertex-set constants $(\mathbf{P}, \mathbf{S})$

Existential MSO$_1$

- Vertex variables (x, y, z)
- Vertex-set variables (X, Y, Z)
- Vertex-set constants $(\mathbf{P}, \mathbf{S})$
- $\exists XYwz$
- $x = y$
- $z \in X$
- $E(x, y)$

Existential MSO_1

- Vertex variables (x, y, z)
- Vertex-set variables (X, Y, Z)
- Vertex-set constants $(\mathbf{P}, \mathbf{S})$
- $\exists XYwz$
- $x = y$
- $z \in X$
- $E(x, y)$
- Negations are only on **quantifier-free formulas**

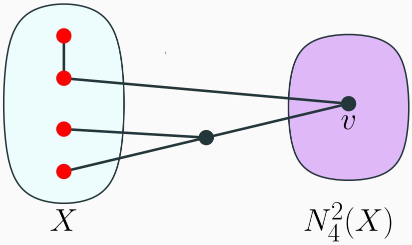
Existential MSO_1

- Vertex variables (x, y, z)
- Vertex-set variables (X, Y, Z)
- Vertex-set constants $(\mathbf{P}, \mathbf{S})$
- $\exists XYwz$
- $x = y$
- $z \in X$
- $E(x, y)$
- Negations are only on **quantifier-free formulas**

$$\exists Xyz \quad (y \in X \lor y = z \lor z \in \mathbf{P}) \lor \neg E(x, y)$$

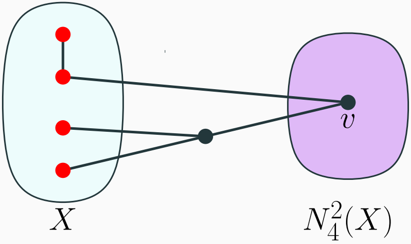
Neighborhood operator

$N_d^r(X)$ is the set of vertices v **at distance at most** r to **at least** d **vertices** in $X \setminus \{v\}$



Neighborhood operator

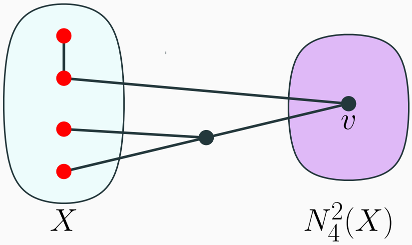
$N_d^r(X)$ is the set of vertices v **at distance at most** r to **at least** d **vertices** in $X \setminus \{v\}$



- $N_1^1(X) = \bigcup_{x \in X} N(x)$

Neighborhood operator

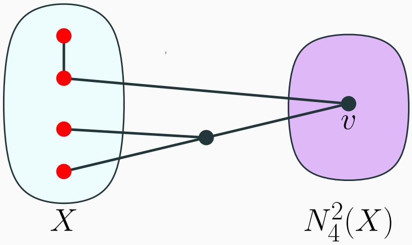
$N_d^r(X)$ is the set of vertices v **at distance at most** r to **at least** d **vertices** in $X \setminus \{v\}$



- $N_1^1(X) = \bigcup_{x \in X} N(x)$
- $N_d^1(X)$ is the set of vertices with at least d neighbors in X

Neighborhood operator

$N_d^r(X)$ is the set of vertices v **at distance at most** r to **at least** d **vertices** in $X \setminus \{v\}$



- $N_1^1(X) = \bigcup_{x \in X} N(x)$
- $N_d^1(X)$ is the set of vertices with at least d neighbors in X
- $N_1^r(\{x\}) = N_{G^r}(x)$.

Neighborhood terms

They are built from:

- Set variables (X, Y, Z) and constants $(\mathbf{P}, \mathbf{S}, \emptyset)$

and **other neighborhoods terms** by applying:

- **Neighborhood operator**: $N_d^r(t)$
- Basic set operations: $t_1 \cap t_2$, $t_1 \cup t_2$, $t_1 \setminus t_2$ and $\bar{t}$

Definition

Distance neighborhood logic (DN)

Extension of **existential $\mathbf{MSO}_1$** with

- **Size measurement of terms**: $|t| \leqslant m$ or $|t| \geq m$
- **Comparison between terms**: $t_1 \subseteq t_2$ or $t_1 = t_2$

Definition

Distance neighborhood logic (DN)

Extension of **existential MSO$_1$** with

- **Size measurement of terms**: $|t| \leqslant m$ or $|t| \geq m$
- **Comparison between terms**: $t_1 \subseteq t_2$ or $t_1 = t_2$

A&C DN

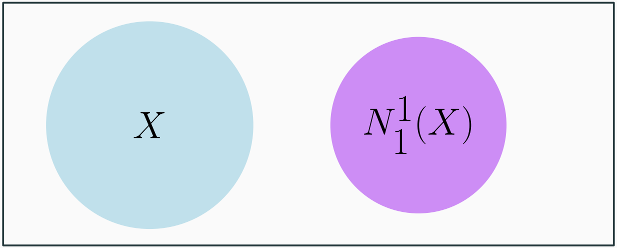
Extension of **DN** with

- **Connectivity constraints**: con(t)
- **Acyclicity constraints**: acy(t)

Examples

Independent Set

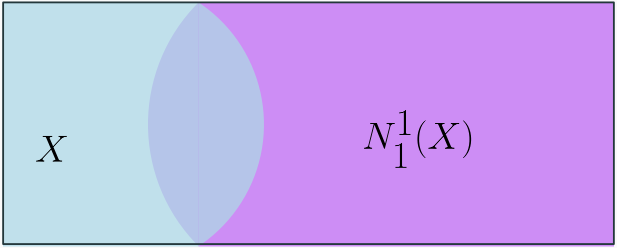
$N_1^1(X) \cap X = \emptyset$



Examples

Dominating Set

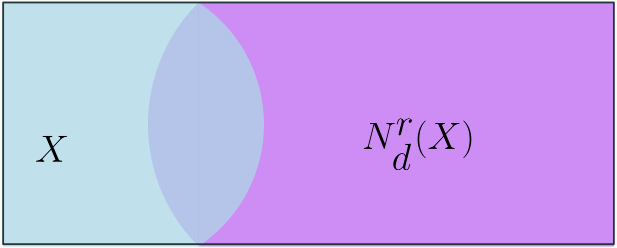
$\overline{N_1^1(X) \cup X} = \emptyset$



Examples

Distance-r d-Dominating Set

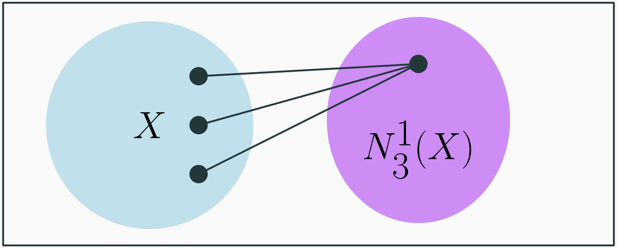
$\overline{N_d^r(X) \cup X} = \emptyset$



Examples

Induced Path

$\mathsf{con}(X) \wedge \mathsf{acy}(X) \wedge X \cap N_3^1(X) = \emptyset$



mim-width	A&C DN	twin-width	FO
clique-width	rank-width		MSO_1
treewidth	branchwidth		MSO_2

Theorem

There is a **model checking** algorithm for A&C DN that runs in time $n^{O(dw|\varphi|^2)}$ where:

- $d = d(\varphi)$ is the **largest value** such that $N_d^{\cdot}(\cdot)$ occurs in φ.
- w the **mim-width** of the given decomposition.

If $r(\varphi) = O(1)$, then the algorithm runs in $n^{O(dw|\varphi|)}$ time.

Other width measures

Theorem

When:

- $r = r(\varphi)$ is the **largest value** such that $N^r_\cdot(\cdot)$ occurs in φ.
- $M = (\prod_{|t_i| \leqslant m_i} m_i)^{O(1)}$.

The **run time** of our algorithm is upper bounded by:

- $2^{O(d(wr|\varphi|)^2)} n^{O(1)} M$ for **tree-width or clique-width**.
- $2^{O(dw^4(r|\varphi|)^2)} n^{O(1)} M$ for **rank-width**.

Better upper bounds when $r = 1$.

Generalization of previous results

Locally Checkable Vertex Subset (LCVS)

INDEPENDENT SET
DOMINATING SET
INDUCED MATCHING
INDUCED d-REGULAR SUBGRAPH
PERFECT CODE
TOTAL DOMINATING SET

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-COLORING
H-HOMOMORPHISM
H-COVERING
ODD CYCLE TRANSVERSAL
PERFECT MATCHING CUT
• • •

[Bui-Xuan, Telle and Vatshelle, 2013]

[Jaffke, Kwon, and Telle, 2018]

FEEDBACK VERTEX SET

Induced Path Problems

INDUCED DISJOINT PATHS
LONGEST INDUCED PATH
H-INDUCED TOPOLOGICAL MINOR

[Jaffke, Kwon, and Telle, 2017]

CONNECTED, ACYCLIC LCVS
CONNECTED, ACYCLIC LCVP

[Bergougnoux and Kante, 2019]

NODE MULTIWAY CUT
SUBSET FEEDBACK VERTEX SET

[Bergougnoux, Papadopoulos and Telle, 2020]

DISTANCE-r LCVS
DISTANCE-r LCVP

[Jaffke et al. 2018]

SEMITOTAL DOMINATING SET

[Galby, Munaro and Ries, 2020]

q-b-COLORING
$[k]$-ROMAN DOMINATION
CONFLICT-FREE q-COLORING

d-STABLE LOCALLY CHECKABLE PROBLEMS

[Gonzales and Mann, 2022]

Generalization of previous results

Locally Checkable Vertex Subset (LCVS)

Independent Set | Induced d-Regular Subgraph
Dominating Set | Perfect Code
Induced Matching | Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring | Odd Cycle Transversal
H-Homomorphism | Perfect Matching Cut
H-Covering | . . .

[Bui-Xuan, Telle and Vatshelle, 2013]

Induced Path Problems

Induced Disjoint Paths | Longest Induced Path
H-Induced Topological Minor

[Jaffke, Kwon, and Telle, 2017]

[Jaffke, Kwon, and Telle, 2018]

Feedback Vertex Set

Connected, Acyclic LCVS

Connected, Acyclic LCVP

[Bergougnoux and Kante, 2019]

A&C DN

$n^{O(w)}$

Node Multiway Cut | Subset Feedback Vertex Set

[Bergougnoux, Papadopoulos and Telle, 2020]

$n^{O(w^2)}$

Distance-r LCVS
Distance-r LCVP

[Jaffke et al. 2018]

Semitotal Dominating Set

[Galby, Munaro and Ries, 2020]

q-b-Coloring | $[k]$-Roman Domination | Conflict-Free q-Coloring

d-stable Locally Checkable problems

[Gonzales and Mann, 2022]

Applications

- Also works for **optimization**!

- Our algorithm is **efficient**, for **many problems**:
    - It is asymptotically as fast as the best known algorithms

Applications

- Also works for **optimization**!

- Our algorithm is **efficient**, for **many problems**:
  - It is asymptotically as fast as the best known algorithms
  - It matches **ETH lower bounds** **tree-width** and **clique-width** and **rank-width** [B., Korhonen and Nederlof, ArXiv]

Applications

- Also works for **optimization**!
- Our algorithm is **efficient**, for **many problems**:
  - It is asymptotically as fast as the best known algorithms
  - It matches **ETH lower bounds** **tree-width** and **clique-width** and **rank-width** [B., Korhonen and Nederlof, ArXiv]
  - Close to the **best-known ETH lower bound** for mim-width: $n^{o(w/\log w)}$ [Bakkane and Jafkke, 2022+]

Applications

- **Solution diversity**: output is q diverse solutions to a given problem Π

Applications

- **Solution diversity**: output is q diverse solutions to a given problem Π

Theorem

If Π is expressible in A&C DN, then **Min-Diverse** Π and **Sum-Diverse** Π are also expressible in A&C DN with "nice" formula.

Applications

- **Solution diversity**: output is q diverse solutions to a given problem Π

Theorem

If Π is expressible in A&C DN, then **Min-Diverse** Π and **Sum-Diverse** Π are also expressible in A&C DN with "nice" formula.

- Variants of q-**Coloring** with fixed number of colors:

Applications

- **Solution diversity**: output is q diverse solutions to a given problem Π

Theorem

If Π is expressible in A&C DN, then **Min-Diverse** Π and **Sum-Diverse** Π are also expressible in A&C DN with "nice" formula.

- Variants of q-**Coloring** with fixed number of colors:
  - b-Coloring, Acyclic Coloring, Star Coloring, Conflict-free colorings...

Applications

- **Solution diversity**: output is q diverse solutions to a given problem Π

Theorem

If Π is expressible in A&C DN, then **Min-Diverse** Π and **Sum-Diverse** Π are also expressible in A&C DN with "nice" formula.

- Variants of q-**Coloring** with fixed number of colors:
  - b-Coloring, Acyclic Coloring, Star Coloring, Conflict-free colorings...
- $L(2,1)$-Labeling and $L(d_1,\dots,d_s)$-Labeling, for fixed number of labels.

Equivalence

Existential Counting Modal Logic (ECML) [Pilipczuk, 2011]

- Vertex-set and **edge-set** variables
- **Modal** = interpretation with an **active vertex** that during evaluation ($\Box\varphi = \varphi$ holds on some neighbors of the active vertex)
- Allows **ultimately periodic counting**

Model-Checking algorithm with running time $2^{O(\mathsf{tw}(G))} \cdot n^{O(1)}$.

Equivalence

Existential Counting Modal Logic (ECML) [Pilipczuk, 2011]

- Vertex-set and **edge-set** variables
- **Modal** = interpretation with an **active vertex** that during evaluation ($\Box\varphi = \varphi$ holds on some neighbors of the active vertex)
- Allows **ultimately periodic counting**

Model-Checking algorithm with running time $2^{O(\mathsf{tw}(G))} \cdot n^{O(1)}$.

Theorem

DN is **equivalent** to the variant of ECML logic with:

- without **edge-set** variables and **ultimately periodic counting**
- **with** access to the r-th power of the input graph

Tightness

Natural extensions of DN

- DN $+ \forall$: we allow a **single universal quantifier**
- DN $+$ EdgeSet: we allow **one edge-set variable** Y and $N_Y(t)$
- DN $+$ Parity: we allow **parity counting** with $N_{\text{even}}(t)$

Theorem

The model checking problems for these extensions are **para-NP-hard** parameterized by formula length plus the (linear) **mim-width** of a given decomposition.

Tightness

Natural extensions of DN

- DN $+ \forall$: we allow a **single universal quantifier**
- DN $+$ EdgeSet: we allow **one edge-set variable** Y and $N_Y(t)$
- DN $+$ Parity: we allow **parity counting** with $N_{\text{even}}(t)$

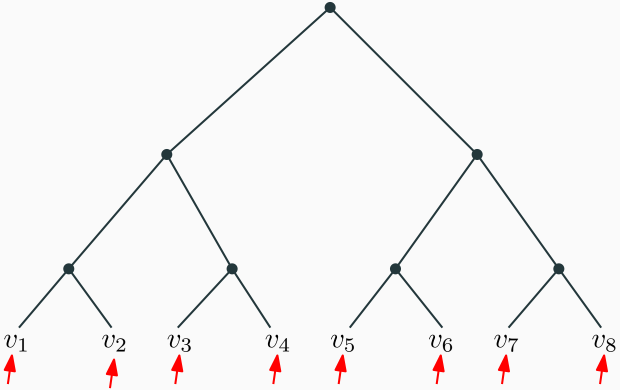
Theorem

The model checking problems for these extensions are **para-NP-hard** parameterized by formula length plus the (linear) **mim-width** of a given decomposition.

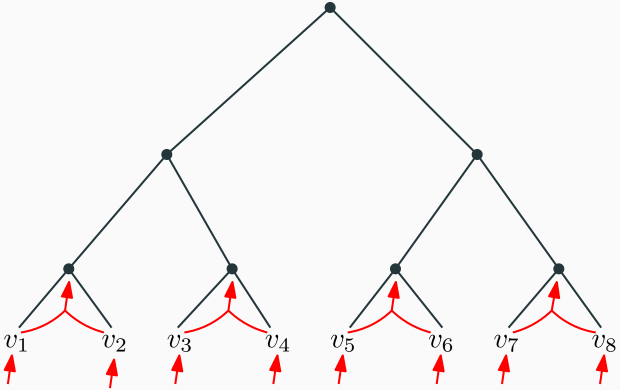
- DN $+ \forall$ can express **Clique**
- DN $+$ EdgeSet and DN $+$ Parity can express problems that are NP-hard on interval graphs (mim-width 1)

Model checking algorithm

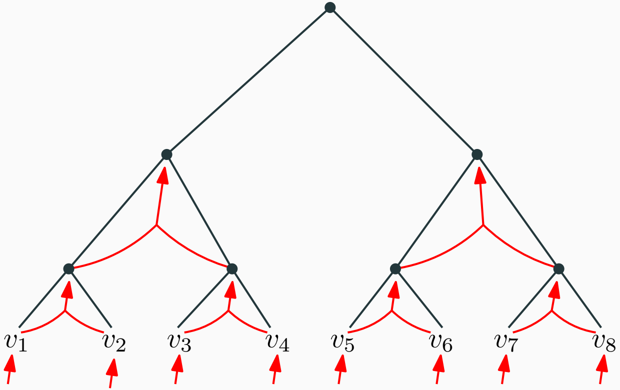
Bottom-up traversal



Bottom-up traversal

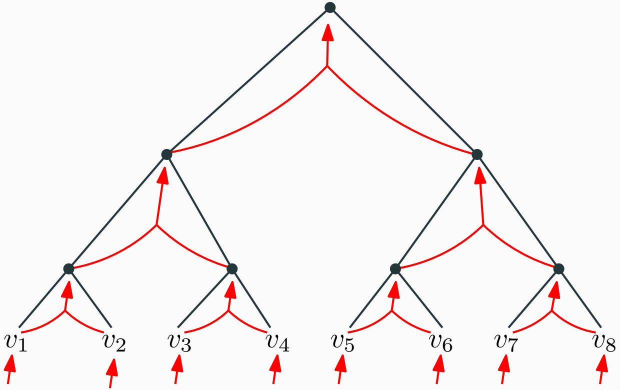


Bottom-up traversal



Bottom-up traversal

At the root: we obtain a solution.



For every cut $(A, \overline{A})$ of the decomposition, we compute a set $\mathcal{B}_A$ of **partial solutions**

- Partial solutions of A are **interpretations of the variables of** φ in $G[A]$.
- Invariant: $\mathcal{B}_A$ **represents all partial solutions of** A
- **Challenge**: keep the size of $\mathcal{B}_A$ **small**

Logic simplifications

Definition

- **Core** DN **Logic**: DN formula with no **vertex variables** and no **neighborhood term** $N_d^r(t)$ where t is not a set variable.

Logic simplifications

Definition

- **Core** DN **Logic**: DN formula with no **vertex variables** and no **neighborhood term** $N_d^r(t)$ where t is not a set variable.
- A&C **clauses**: A&C DN formula that are **conjunctions** of a core DN formula and predicates acy(X) or con(X).

$$\exists X, Y \quad X \cap N_1^1(X) = \emptyset \wedge Y \subseteq N_1^1(X) \wedge \mathsf{acy}(Y) \wedge \mathsf{con}(Y).$$

Logic simplifications

Definition

- **Core** DN **Logic**: DN formula with no **vertex variables** and no **neighborhood term** $N_d^r(t)$ where t is not a set variable.
- A&C **clauses**: A&C DN formula that are **conjunctions** of a core DN formula and predicates $\mathsf{acy}(X)$ or $\mathsf{con}(X)$.

$$\exists X, Y \quad X \cap N_1^1(X) = \emptyset \wedge Y \subseteq N_1^1(X) \wedge \mathsf{acy}(Y) \wedge \mathsf{con}(Y).$$

Theorem (Very informal)

It is **sufficient** to prove the meta-theorem for A&C **clauses.**

Crucial tool

Definition

Two partial solutions $\tilde{B}, \tilde{C}$ of A are **φ-equivalent** over A ($\tilde{B} \equiv_{\varphi}^{A} \tilde{C}$) if $N_d^r(\tilde{B}(X)) \cap \overline{A} = N_d^r(\tilde{C}(X)) \cap \overline{A}$ for **every $N_d^r(X)$ occurring in φ.**

Lemma

The number $\mathrm{nec}_{\varphi}(A)$ of **equivalence classes** of $\equiv_{\varphi}^{A}$ is upper bounded by

- $n^{O(dw|\varphi|)}$ for **mim-width**
- $2^{O(d(wr)^2|\varphi|)}$ for **tree-width or clique-width**
- $2^{O(dw^4r^2|\varphi|)}$ for **rank-width**

$\equiv^A_\varphi$ is a generalization of the d-**neighbor equivalence relation** from [Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Subset (LCVS)

INDEPENDENT SET
DOMINATING SET
INDUCED MATCHING
INDUCED d-REGULAR SUBGRAPH
PERFECT CODE
TOTAL DOMINATING SET

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-COLORING
H-HOMOMORPHISM
H-COVERING
ODD CYCLE TRANSVERSAL
PERFECT MATCHING CUT
...

[Bui-Xuan, Telle and Vatshelle, 2013]

[Jaffke, Kwon, and Telle, 2018]

FEEDBACK VERTEX SET

Induced Path Problems

INDUCED DISJOINT PATHS
LONGEST INDUCED PATH
H-INDUCED TOPOLOGICAL MINOR

[Jaffke, Kwon, and Telle, 2017]

CONNECTED, ACYCLIC LCVS

CONNECTED, ACYCLIC LCVP

[Bergougnoux and Kante, 2019]

NODE MULTIWAY CUT
SUBSET FEEDBACK VERTEX SET

[Bergougnoux, Papadopoulos and Telle, 2020]

DISTANCE-r LCVS
DISTANCE-r LCVP

[Jaffke et al. 2018]

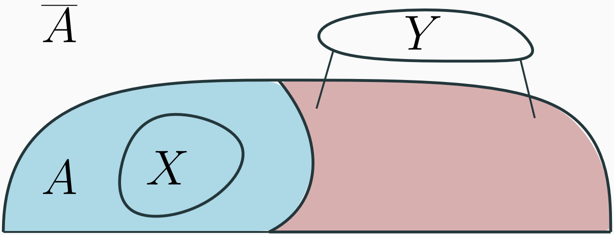
SEMITOTAL DOMINATING SET
[Galby, Munaro and Ries, 2020]

q-b-COLORING
$[k]$-ROMAN DOMINATION
CONFLICT-FREE q-COLORING
d-STABLE LOCALLY CHECKABLE PROBLEMS

[Gonzales and Mann, 2022]

Expectation

Intuition: We cannot classify/compare **partial solutions** from $G[A]$ without knowing how they will be **completed by** $\overline{A}$.



Definition

An expectation $\mathbb{E}$ over A is an **equivalence class** of $\equiv^{\varphi}_{A}$.

Dealing with core DN

Lemma (informal)

For every core DN formula φ and **expectation** $\mathbb{E}$ over A, there exists a **"nice" equivalence relation** $\bowtie_{\mathbb{E}}$ such that:

- for all partial solutions $\tilde{B}, \tilde{C}$ of A such that $\tilde{B} \bowtie_{\mathbb{E}} \tilde{C}$
- for all partial solutions $\tilde{D}$ of $\overline{A}$ in $\mathbb{E}$

$\tilde{B} \cup \tilde{D}$ is a solution **iff** $\tilde{C} \cup \tilde{D}$ is a solution.

Dealing with core DN

Lemma (informal)

For every core DN formula φ and **expectation** $\mathbb{E}$ over A, there exists a **"nice" equivalence relation** $\bowtie_{\mathbb{E}}$ such that:

- for all partial solutions $\tilde{B}, \tilde{C}$ of A such that $\tilde{B} \bowtie_{\mathbb{E}} \tilde{C}$
- for all partial solutions $\tilde{D}$ of $\overline{A}$ in $\mathbb{E}$

$\tilde{B} \cup \tilde{D}$ is a solution **iff** $\tilde{C} \cup \tilde{D}$ is a solution.

- Enough to prove the meta-theorem for DN
- $\bowtie_{\mathbb{E}}$ refines $\equiv_A^{\varphi}$

Dealing with connectivity and acyclicity

- We **generalize** the tools designed in [B. and Kanté, 2019] based on the **rank-based approach** from [Bodlaender et al., 2013]

Locally Checkable Vertex Subset (LCVS)

INDEPENDENT SET — INDUCED d-REGULAR SUBGRAPH
DOMINATING SET — PERFECT CODE
INDUCED MATCHING — TOTAL DOMINATING SET

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-COLORING — ODD CYCLE TRANSVERSAL
H-HOMOMORPHISM — PERFECT MATCHING CUT
H-COVERING — • • •

[Bui-Xuan, Telle and Vatshelle, 2013]

Induced Path Problems

INDUCED DISJOINT PATHS — LONGEST INDUCED PATH
H-INDUCED TOPOLOGICAL MINOR

[Jaffke, Kwon, and Telle, 2017]

[Jaffke, Kwon, and Telle, 2018]
FEEDBACK VERTEX SET

CONNECTED, ACYCLIC LCVS
CONNECTED, ACYCLIC LCVP

[Bergougnoux and Kante, 2019]

NODE MULTIWAY CUT — SUBSET FEEDBACK VERTEX SET

[Bergougnoux, Papadopoulos and Telle, 2020]

DISTANCE-r LCVS
DISTANCE-r LCVP

[Jaffke et al. 2018]

SEMITOTAL DOMINATING SET
[Galby, Munaro and Ries, 2020]

q-b-COLORING — $[k]$-ROMAN DOMINATION — CONFLICT-FREE q-COLORING
d-STABLE LOCALLY CHECKABLE PROBLEMS

[Gonzales and Mann, 2022]

Conclusion

Conclusion

mim-width A&C DN		twin-width FO
clique-width	rank-width	MSO_1
treewidth	branchwidth	MSO_2

- Efficient model checking algorithms for several width parameters and **tight** in several ways
- Work in progress: capturing Subset FVS and **Subset OCT**

Open questions

- **Algorithmic applications**: Tight lower bounds for **rank-width** and **mim-width**!

- **Complexity of computing a good decomposition**: tough open questions for **mim-width** and **twin-width**!

Computing mim-width

[Saether and Vatshelle, 2015]

Computing **mim-width** is $W[1]$-hard and there is no $O(1)$-approximation in polytime unless NP = ZPP.

- Can we recognize (**strongly chordal**) graphs of **mim-width 1** in polytime?
- Can we recognize **leaf powers** in polytime?

Thank you