Tight Complexity Bounds for Counting Generalized Dominating Sets in Bounded-Treewidth Graphs

Jacob Focke[1] Dániel Marx[1] Fionn Mc Inerney[1] Daniel Neuen[2]
Govind S. Sankar[3] **Philipp Schepper[1]** Philip Wellnitz[4]

[1] CISPA Helmholtz Center for Information Security
[2] Simon Fraser University
[3] Duke University
[4] MPI Informatics, SIC

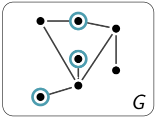
January 24, 2023

Counting Vertex Selections on Graphs with Small Treewidth

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

#Independent Set

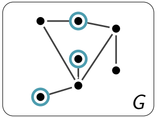
Count size-k subsets $S \subseteq V(G)$
of pairwise nonadjacent vertices.



Counting Vertex Selections on Graphs with Small Treewidth

#Independent Set

Count size-k subsets $S \subseteq V(G)$ of pairwise nonadjacent vertices.

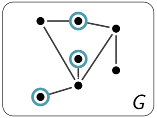


- $2^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm [N'06]
  given a tree decomposition of width $\mathrm{tw}(G)$
- No $(2-\varepsilon)^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm (under SETH) [LMS'18]

Counting Vertex Selections on Graphs with Small Treewidth

#Independent Set

Count size-k subsets $S \subseteq V(G)$
of pairwise nonadjacent vertices.



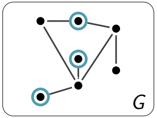
- $2^{\text{tw}(G)} \cdot \text{poly}(|G|)$ algorithm [N'06]
  given a tree decomposition of width $\text{tw}(G)$
- No $(2-\varepsilon)^{\text{tw}(G)} \cdot \text{poly}(|G|)$ algorithm
  (under SETH) [LMS'18]

Strong Exponential Time Hypothesis (SETH):
For any $\delta > 0$, there is a large enough k such that there is no $(2-\delta)^n$ algorithm for k-CNF-SAT.

Counting Vertex Selections on Graphs with Small Treewidth

#Independent Set

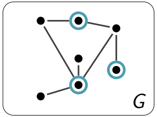
Count size-k subsets $S \subseteq V(G)$
of pairwise nonadjacent vertices.



- $2^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm [N'06]
  given a tree decomposition of width $\mathrm{tw}(G)$
- No $(2-\varepsilon)^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm
  (under SETH) [LMS'18]

#Dominating Set

Count size-k subsets $S \subseteq V(G)$
whose neighborhoods cover $V(G) \setminus S$.

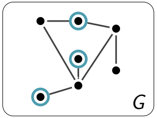


Strong Exponential Time Hypothesis (SETH):
For any $\delta > 0$, there is a large enough k such that there is no $(2-\delta)^n$ algorithm for k-CNF-SAT.

Counting Vertex Selections on Graphs with Small Treewidth

#Independent Set

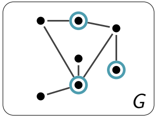
Count size-k subsets $S \subseteq V(G)$
of pairwise nonadjacent vertices.



- $2^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm [N'06]
  given a tree decomposition of width $\mathrm{tw}(G)$
- No $(2-\varepsilon)^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm (under SETH) [LMS'18]

#Dominating Set

Count size-k subsets $S \subseteq V(G)$
whose neighborhoods cover $V(G) \setminus S$.



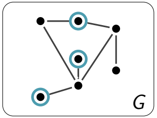
- $3^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm [vRBR'09]
  given a tree decomposition of width $\mathrm{tw}(G)$
- No $(3-\varepsilon)^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ algorithm (under SETH) [LMS'18]

Strong Exponential Time Hypothesis (SETH):
For any $\delta > 0$, there is a large enough k such that there is no $(2-\delta)^n$ algorithm for k-CNF-SAT.

A Generalized Problem

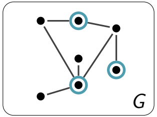
#Independent Set

Count size-k subsets $S \subseteq V(G)$
of pairwise nonadjacent vertices.



#Dominating Set

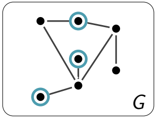
Count size-k subsets $S \subseteq V(G)$
whose neighborhoods cover $V(G) \setminus S$.



A Generalized Problem

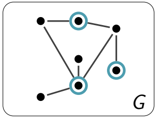
#Independent Set

Count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \{0\}$
and for all $s \notin S$: $|N(s) \cap S| \in \mathbb{N}$.



#Dominating Set

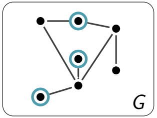
Count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \mathbb{N}$
and for all $s \notin S$: $|N(s) \cap S| \in \mathbb{N} \setminus \{0\}$.



A Generalized Problem

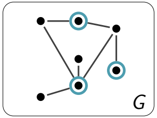
#Independent Set

Count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \{0\}$
and for all $s \notin S$: $|N(s) \cap S| \in \mathbb{N}$.



#Dominating Set

Count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \mathbb{N}$
and for all $s \notin S$: $|N(s) \cap S| \in \mathbb{N} \setminus \{0\}$.



#(σ, ρ)-Dominating Set for fixed $\sigma, \rho \subseteq \mathbb{N}$ [Telle'94]

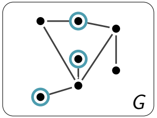
Given a graph G, count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \sigma$,
for all $s \notin S$: $|N(s) \cap S| \in \rho$.

We consider finite or cofinite σ and ρ.

A Generalized Problem

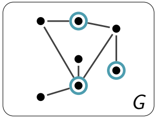
$\#(\{0\}, \mathbb{N})$-Dominating Set

Count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \{0\}$
and for all $s \notin S$: $|N(s) \cap S| \in \mathbb{N}$.



$\#(\mathbb{N}, \mathbb{N} \setminus \{0\})$-Dominating Set

Count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \mathbb{N}$
and for all $s \notin S$: $|N(s) \cap S| \in \mathbb{N} \setminus \{0\}$.



$\#(\sigma, \rho)$-Dominating Set for fixed $\sigma, \rho \subseteq \mathbb{N}$ [Telle'94]

Given a graph G, count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \sigma$,
for all $s \notin S$: $|N(s) \cap S| \in \rho$.

We consider finite or cofinite σ and ρ.

$\#(\sigma,\rho)$-Dominating Set Is Relevant

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

$\#(\sigma,\rho)$-Dominating Set for fixed $\sigma, \rho \subseteq \mathbb{N}$ [Telle'94]

Given a graph G, count $S \subseteq V(G)$ with $|S| = k$ s.t.
for all $s \in S$: $|N(s) \cap S| \in \sigma$ and for all $s \notin S$: $|N(s) \cap S| \in \rho$.

This generalizes many well-known problems including

Classical name	σ	ρ
Independent Set	$\{0\}$	$\mathbb{N}$
Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$
Strong Independent Set	$\{0\}$	$\{0,1\}$
Independent Dominating Set	$\{0\}$	$\mathbb{N} \setminus \{0\}$
Perfect Code/Exact Independent Dominating Set	$\{0\}$	$\{1\}$
Total Dominating Set	$\mathbb{N} \setminus \{0\}$	$\mathbb{N} \setminus \{0\}$
Perfect Dominating Set	$\mathbb{N}$	$\{1\}$
Induced Bounded-Degree Subgraph	$\{0,1,\ldots,d\}$	$\mathbb{N}$
Induced d-Regular Subgraph	$\{d\}$	$\mathbb{N}$

About Partial Solutions

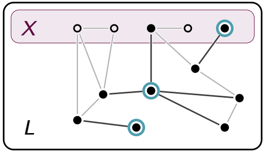
Consider finite $\sigma = \{0, 2\}$ and cofinite $\rho = \{1, 4, 5, 6, 7, \dots\} = \mathbb{N} \setminus \{0, 2, 3\}$

A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L

About Partial Solutions

Consider finite $\sigma = \{0, 2\}$ and cofinite $\rho = \{1, 4, 5, 6, 7, \dots\} = \mathbb{N} \setminus \{0, 2, 3\}$

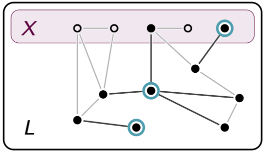
A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L



About Partial Solutions

Consider finite $\sigma = \{0, 2\}$ and cofinite $\rho = \{1, 4, 5, 6, 7, \ldots\} = \mathbb{N} \setminus \{0, 2, 3\}$

A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L



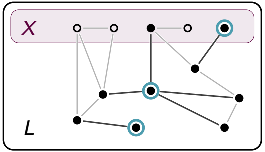
Consider a *selected* vertex v in $X \cap S$:

- > 2 neighbors in S? S is invalid!

About Partial Solutions

Consider finite $\sigma = \{0, \mathbf{2}\}$ and cofinite $\rho = \{1, 4, 5, 6, 7, \ldots\} = \mathbb{N} \setminus \{0, 2, 3\}$

A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L



Consider a *selected* vertex v in $X \cap S$:

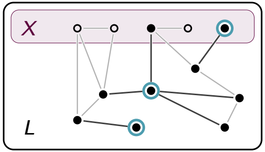
- > 2 neighbors in S? S is invalid!
- $0, 1,$ or 2 neighbors possible

$\rightsquigarrow$ $3 = \max \sigma + 1$ states

About Partial Solutions

Consider finite $\sigma = \{0, \mathbf{2}\}$ and cofinite $\rho = \{1, 4, 5, 6, 7, \dots\} = \mathbb{N} \setminus \{0, 2, 3\}$

A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L



Consider an *unselected* vertex in $X \setminus S$:

- No bound for number of neighbors

Consider a *selected* vertex v in $X \cap S$:

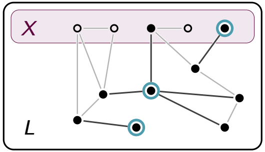
- > 2 neighbors in S? S is invalid!
- $0, 1,$ or 2 neighbors possible

$\rightsquigarrow$ $3 = \max \sigma + 1$ states

About Partial Solutions

Consider finite $\sigma = \{0, \mathbf{2}\}$ and cofinite $\rho = \{1, 4, 5, 6, 7, \dots\} = \mathbb{N} \setminus \{0, 2, \mathbf{3}\}$

A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L



Consider a *selected* vertex v in $X \cap S$:

- > 2 neighbors in S? S is invalid!
- $0, 1$, or 2 neighbors possible

$\rightsquigarrow$ $3 = \max \sigma + 1$ states

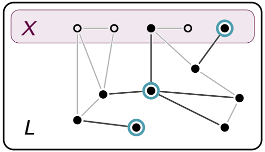
Consider an *unselected* vertex in $X \setminus S$:

- No bound for number of neighbors
- Last "bad" state is "3 neighbors";
  Cannot distinguish $4, 5, \dots$ neighbors
  (v does not care if it gets further neighbors)

About Partial Solutions

Consider finite $\sigma = \{0, \mathbf{2}\}$ and cofinite $\rho = \{1, \mathbf{4}, 5, 6, 7, \dots\} = \mathbb{N} \setminus \{0, 2, \mathbf{3}\}$

A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L



Consider a *selected* vertex v in $X \cap S$:

- > 2 neighbors in S? S is invalid!
- $0, 1$, or 2 neighbors possible

$\rightsquigarrow$ $3 = \max \sigma + 1$ states

Consider an *unselected* vertex in $X \setminus S$:

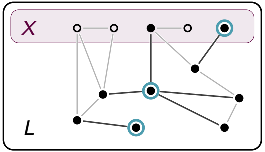
- No bound for number of neighbors
- Last "bad" state is "3 neighbors";
  Cannot distinguish $4, 5, \dots$ neighbors
  (v does not care if it gets further neighbors)
- $0, 1, 2, 3$, or "≥ 4" neighbors possible

$\rightsquigarrow$ $5 = \max(\mathbb{Z} \setminus \rho) + 2$ states

About Partial Solutions

Consider finite $\sigma = \{0, \mathbf{2}\}$ and cofinite $\rho = \{1, \mathbf{4}, 5, 6, 7, \dots\} = \mathbb{N} \setminus \{0, 2, \mathbf{3}\}$

A separator X, a side $L \supseteq X$ of the separated graph, a partial solution S for L



Consider a *selected* vertex v in $X \cap S$:

- > 2 neighbors in S? S is invalid!
- $0, 1,$ or 2 neighbors possible

$\rightsquigarrow$ $3 = \max\sigma + 1$ states

Consider an *unselected* vertex in $X \setminus S$:

- No bound for number of neighbors
- Last "bad" state is "3 neighbors";
  Cannot distinguish $4, 5, \dots$ neighbors
  (v does not care if it gets further neighbors)
- $0, 1, 2, 3,$ or "≥ 4" neighbors possible

$\rightsquigarrow$ $5 = \max(\mathbb{Z} \setminus \rho) + 2$ states

Consider $(3 + 5)^{|X|} = ((\max\sigma + 1) + (\max(\mathbb{Z} \setminus \rho) + 2))^{|X|}$ states for the vertices in X.

Solving #(σ, ρ)-Dominating Set

$$\sigma_{\text{top}} := \begin{cases} \max \sigma & \text{for finite } \sigma; \\ 1 + \max(\mathbb{Z} \setminus \sigma) & \text{for cofinite } \sigma. \end{cases} \qquad \rho_{\text{top}} := \begin{cases} \max \rho & \text{for finite } \rho; \\ 1 + \max(\mathbb{Z} \setminus \rho) & \text{for cofinite } \rho. \end{cases}$$

Solving $\#(\sigma, \rho)$-Dominating Set

$$\sigma_{\text{top}} := \begin{cases} \max \sigma & \text{for finite } \sigma; \\ 1 + \max(\mathbb{Z} \setminus \sigma) & \text{for cofinite } \sigma. \end{cases} \qquad \rho_{\text{top}} := \begin{cases} \max \rho & \text{for finite } \rho; \\ 1 + \max(\mathbb{Z} \setminus \rho) & \text{for cofinite } \rho. \end{cases}$$

Previous observations and fast convolution techniques give:

Theorem (van Rooij'20)

$\#(\sigma, \rho)$-Dominating Set *can be solved in time* $(\sigma_{\text{top}} + \rho_{\text{top}} + 2)^{\text{tw}(G)} \cdot \text{poly}(|G|)$ *if* σ, ρ *are finite or cofinite and a tree decomposition of width* $\text{tw}(G)$ *is given.*

Solving #(σ, ρ)-Dominating Set

$$\sigma_{\text{top}} := \begin{cases} \max \sigma & \text{for finite } \sigma; \\ 1 + \max(\mathbb{Z} \setminus \sigma) & \text{for cofinite } \sigma. \end{cases} \qquad \rho_{\text{top}} := \begin{cases} \max \rho & \text{for finite } \rho; \\ 1 + \max(\mathbb{Z} \setminus \rho) & \text{for cofinite } \rho. \end{cases}$$

Previous observations and fast convolution techniques give:

Theorem (van Rooij'20)

#(σ, ρ)-Dominating Set *can be solved in time* $(\sigma_{\text{top}} + \rho_{\text{top}} + 2)^{\text{tw}(G)} \cdot \text{poly}(|G|)$ *if* σ, ρ *are finite or cofinite and a tree decomposition of width* tw(G) *is given.*

	σ	ρ	Runtime (ignoring polynomial terms)
#Independent Set	$\{0\}$	$\mathbb{N}$	$(0+0+2)^{\text{tw}} = 2^{\text{tw}}$
#Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$	$(0+1+2)^{\text{tw}} = 3^{\text{tw}}$
#Ind d-reg Subgraph	$\{d\}$	$\mathbb{N}$	$(d+0+2)^{\text{tw}} = (d+2)^{\text{tw}}$
#Perfect Code	$\{0\}$	$\{1\}$	$(0+1+2)^{\text{tw}} = 3^{tw}$

Solving $\#(\sigma, \rho)$-Dominating Set

$$\sigma_{\text{top}} := \begin{cases} \max \sigma & \text{for finite } \sigma; \\ 1 + \max(\mathbb{Z} \setminus \sigma) & \text{for cofinite } \sigma. \end{cases} \qquad \rho_{\text{top}} := \begin{cases} \max \rho & \text{for finite } \rho; \\ 1 + \max(\mathbb{Z} \setminus \rho) & \text{for cofinite } \rho. \end{cases}$$

Previous observations and fast convolution techniques give:

Theorem (van Rooij'20)

$\#(\sigma, \rho)$-Dominating Set *can be solved in time* $(\sigma_{\text{top}} + \rho_{\text{top}} + 2)^{\text{tw}(G)} \cdot \text{poly}(|G|)$
if σ, ρ *are finite or cofinite and a tree decomposition of width* $\text{tw}(G)$ *is given.*

	σ	ρ	Runtime (ignoring polynomial terms)	
#Independent Set	$\{0\}$	$\mathbb{N}$	$(0+0+2)^{\text{tw}} = 2^{\text{tw}}$	tight
#Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$	$(0+1+2)^{\text{tw}} = 3^{\text{tw}}$	tight
#Ind d-reg Subgraph	$\{d\}$	$\mathbb{N}$	$(d+0+2)^{\text{tw}} = (d+2)^{\text{tw}}$	
#Perfect Code	$\{0\}$	$\{1\}$	$(0+1+2)^{\text{tw}} = 3^{tw}$	

Solving #(σ, ρ)-Dominating Set

$$\sigma_{\text{top}} := \begin{cases} \max \sigma & \text{for finite } \sigma; \\ 1 + \max(\mathbb{Z} \setminus \sigma) & \text{for cofinite } \sigma. \end{cases} \qquad \rho_{\text{top}} := \begin{cases} \max \rho & \text{for finite } \rho; \\ 1 + \max(\mathbb{Z} \setminus \rho) & \text{for cofinite } \rho. \end{cases}$$

Previous observations and fast convolution techniques give:

Theorem (van Rooij'20)

#(σ, ρ)-Dominating Set can be solved in time $(\sigma_{\text{top}} + \rho_{\text{top}} + 2)^{\text{tw}(G)} \cdot \text{poly}(|G|)$ if σ, ρ are finite or cofinite and a tree decomposition of width $\text{tw}(G)$ is given.

	σ	ρ	Runtime (ignoring polynomial terms)	
#Independent Set	$\{0\}$	$\mathbb{N}$	$(0+0+2)^{\text{tw}} = 2^{\text{tw}}$	tight
#Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$	$(0+1+2)^{\text{tw}} = 3^{\text{tw}}$	tight
#Ind d-reg Subgraph	$\{d\}$	$\mathbb{N}$	$(d+0+2)^{\text{tw}} = (d+2)^{\text{tw}}$	?
#Perfect Code	$\{0\}$	$\{1\}$	$(0+1+2)^{\text{tw}} = 3^{tw}$	?

Question: Is this algorithm optimal?

Solving #(σ, ρ)-Dominating Set

$$\sigma_{\text{top}} := \begin{cases} \max \sigma & \text{for finite } \sigma; \\ 1 + \max(\mathbb{Z} \setminus \sigma) & \text{for cofinite } \sigma. \end{cases} \qquad \rho_{\text{top}} := \begin{cases} \max \rho & \text{for finite } \rho; \\ 1 + \max(\mathbb{Z} \setminus \rho) & \text{for cofinite } \rho. \end{cases}$$

Previous observations and fast convolution techniques give:

Theorem (van Rooij'20)

#(σ, ρ)-Dominating Set *can be solved in time* $(\sigma_{\text{top}} + \rho_{\text{top}} + 2)^{\text{tw}(G)} \cdot \text{poly}(|G|)$
if σ, ρ *are finite or cofinite and a tree decomposition of width* $\text{tw}(G)$ *is given.*

	σ	ρ	Runtime (ignoring polynomial terms)	
#Independent Set	$\{0\}$	$\mathbb{N}$	$(0+0+2)^{\text{tw}} = 2^{\text{tw}}$	tight
#Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$	$(0+1+2)^{\text{tw}} = 3^{\text{tw}}$	tight
#Ind d-reg Subgraph	$\{d\}$	$\mathbb{N}$	$(d+0+2)^{\text{tw}} = (d+2)^{\text{tw}}$	?
#Perfect Code	$\{0\}$	$\{1\}$	$(0+1+2)^{\text{tw}} = 3^{tw}$	?

Question: Is this algorithm optimal?

No! We improve it for many (σ, ρ) and show that our improvement is optimal.

Structured Pairs

m-structured (σ, ρ)

For m ≥ 2, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \mod m$ and for all $r \in \rho$ we have $r \equiv \beta \mod m$.

Structured Pairs

m-structured (σ, ρ)

For $m \geq 2$, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \mod m$ and for all $r \in \rho$ we have $r \equiv \beta \mod m$.

Examples:

σ	ρ	m-structured for
$\{1, 3\}$	$\{4\}$	$m = 2$
$\{0, 4\}$	$\{1, 9\}$	
$\{0\}$	$\{1\}$	
$\{0, 3\}$	$\{1, 5\}$	
$\{d\}$	$\mathbb{N}$	

Structured Pairs

m-structured (σ, ρ)

For m ≥ 2, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha$ mod m and for all $r \in \rho$ we have $r \equiv \beta$ mod m.

Examples:

σ	ρ	m-structured for
$\{1, 3\}$	$\{4\}$	m $= 2$
$\{0, 4\}$	$\{1, 9\}$	m $= 2, 4$
$\{0\}$	$\{1\}$	
$\{0, 3\}$	$\{1, 5\}$	
$\{d\}$	$\mathbb{N}$	

Structured Pairs

m-structured (σ, ρ)

For $m \geq 2$, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \bmod m$ and for all $r \in \rho$ we have $r \equiv \beta \bmod m$.

Examples:

σ	ρ	m-structured for
$\{1, 3\}$	$\{4\}$	$m = 2$
$\{0, 4\}$	$\{1, 9\}$	$m = 2, 4$
$\{0\}$	$\{1\}$	every $m \geq 2$
$\{0, 3\}$	$\{1, 5\}$	
$\{d\}$	$\mathbb{N}$	

Structured Pairs

m-structured (σ, ρ)

For $m \geq 2$, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \bmod m$ and for all $r \in \rho$ we have $r \equiv \beta \bmod m$.

Examples:

σ	ρ	m-structured for
$\{1,3\}$	$\{4\}$	$m = 2$
$\{0,4\}$	$\{1,9\}$	$m = 2, 4$
$\{0\}$	$\{1\}$	every $m \geq 2$
$\{0,3\}$	$\{1,5\}$	no $m \geq 2$
$\{d\}$	$\mathbb{N}$	no $m \geq 2$

Structured Pairs

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

m-structured (σ, ρ)

For $m \geq 2$, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \bmod m$ and for all $r \in \rho$ we have $r \equiv \beta \bmod m$.

Examples:

σ	ρ	m-structured for
$\{1, 3\}$	$\{4\}$	$m = 2$
$\{0, 4\}$	$\{1, 9\}$	$m = 2, 4$
$\{0\}$	$\{1\}$	every $m \geq 2$
$\{0, 3\}$	$\{1, 5\}$	no $m \geq 2$
$\{d\}$	$\mathbb{N}$	no $m \geq 2$

If (σ, ρ) is m-structured for some $m \geq 2$, we can get an improved algorithm.

Our Contribution

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not m $\geq$ 3-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

Our Contribution

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{top} + \rho_{top} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 2$ if $\sigma_{top} = \rho_{top}$ is even and (σ, ρ) is 2-structured (but not $m \geq 3$-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 1$ otherwise.

Theorem (Upper Bounds)

$\#(\sigma, \rho)$-DOMINATING SET *can be solved in time* $(c_{\sigma,\rho})^{\mathrm{tw}(G)} \cdot \mathrm{poly}(|G|)$ *if* σ, ρ *are finite or cofinite and a tree decomposition of width* $\mathrm{tw}(G)$ *is given.*

Our Contribution

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not m $\geq$ 3-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

Theorem (Upper Bounds)

#(σ, ρ)-Dominating Set *can be solved in time* $(c_{\sigma,\rho})^{\text{tw}(G)} \cdot \text{poly}(|G|)$ *if* σ, ρ *are finite or cofinite and a tree decomposition of width* $\text{tw}(G)$ *is given.*

Theorem (Lower Bounds)

Unless #SETH fails, #(σ, ρ)-Dominating Set *has no* $(c_{\sigma,\rho} - \varepsilon)^{\text{tw}(G)} \cdot \text{poly}(|G|)$ *algo for non-trivial, finite or cofinite* σ, ρ*, even if a tree decomposition of width* $\text{tw}(G)$ *is given.*

Implications of the New Results

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not $m \geq 3$-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

	σ	ρ	m-structured for	Runtime (ignoring polynomial terms)	
#Independent Set	$\{0\}$	$\mathbb{N}$	no $m \geq 2$	2^{tw}	tight
#Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$	no $m \geq 2$	3^{tw}	tight
#Ind d-reg Subgraph	$\{d\}$	$\mathbb{N}$	no $m \geq 2$	$(d+2)^{\text{tw}}$	?
#Perfect Code	$\{0\}$	$\{1\}$	every $m \geq 2$	3^{tw}	?

Implications of the New Results

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not $m \geq 3$-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

	σ	ρ	m-structured for	Runtime (ignoring polynomial terms)	
#Independent Set	$\{0\}$	$\mathbb{N}$	no $m \geq 2$	2^{tw}	tight
#Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$	no $m \geq 2$	3^{tw}	tight
#Ind d-reg Subgraph	$\{d\}$	$\mathbb{N}$	no $m \geq 2$	$(d+2)^{\text{tw}}$	tight
#Perfect Code	$\{0\}$	$\{1\}$	every $m \geq 2$	3^{tw}	?

Implications of the New Results

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{top} + \rho_{top} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 2$ if $\sigma_{top} = \rho_{top}$ is even and (σ, ρ) is 2-structured (but not $m \geq 3$-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 1$ otherwise.

	σ	ρ	m-structured for	Runtime (ignoring polynomial terms)	
#Independent Set	$\{0\}$	$\mathbb{N}$	no $m \geq 2$	2^{tw}	tight
#Dominating Set	$\mathbb{N}$	$\mathbb{N} \setminus \{0\}$	no $m \geq 2$	3^{tw}	tight
#Ind d-reg Subgraph	$\{d\}$	$\mathbb{N}$	no $m \geq 2$	$(d+2)^{tw}$	tight
#Perfect Code	$\{0\}$	$\{1\}$	every $m \geq 2$	~~3^{tw}~~ 2^{tw}	tight

Upper Bound

m-structured (σ, ρ)

For m ≥ 2, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \bmod \mathrm{m}$ and for all $r \in \rho$ we have $r \equiv \beta \bmod \mathrm{m}$.

Note: Cofinite sets are not m-structured. $\rightsquigarrow$ Only finite sets are relevant.

Upper Bound

m-structured (σ, ρ)

For $m \geq 2$, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \bmod m$ and for all $r \in \rho$ we have $r \equiv \beta \bmod m$.

Note: Cofinite sets are not m-structured. $\rightsquigarrow$ Only finite sets are relevant.

Goal: Get a better bound for the number of states of (partial) solutions.

Upper Bound

m-structured (σ, ρ)

For $m \geq 2$, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \bmod m$ and for all $r \in \rho$ we have $r \equiv \beta \bmod m$.

Note: Cofinite sets are not m-structured. $\rightsquigarrow$ Only finite sets are relevant.

Goal: Get a better bound for the number of states of (partial) solutions.

Then the standard dynamic program can be improved:

- Only a reduced number of states has to be considered

Upper Bound

m-structured (σ, ρ)

For $m \geq 2$, (σ, ρ) is m-structured if there are α and β such that
for all $s \in \sigma$ we have $s \equiv \alpha \bmod m$ and for all $r \in \rho$ we have $r \equiv \beta \bmod m$.

Note: Cofinite sets are not m-structured. $\rightsquigarrow$ Only finite sets are relevant.

Goal: Get a better bound for the number of states of (partial) solutions.

Then the standard dynamic program can be improved:

- Only a reduced number of states has to be considered
- Existing convolution techniques can be extended to handle join nodes efficiently

A Better Bound for the Number of States

Recall: Each vertex can have $\max \sigma + \max \rho + 2$ states. (σ, ρ are finite!)

$\rightsquigarrow$ For a set of k vertices, $(\max \sigma + \max \rho + 2)^k$ combinations must be to considered

A Better Bound for the Number of States

Recall: Each vertex can have $\max \sigma + \max \rho + 2$ states. (σ, ρ are finite!)
$\rightsquigarrow$ For a set of k vertices, $(\max \sigma + \max \rho + 2)^k$ combinations must be to considered

Assume (σ, ρ) is m-structured (e.g., $m = 2$):

- The selection status of specific vertices determines the number of neighbors for other vertices (think of it as some parity-like argument).

A Better Bound for the Number of States

Recall: Each vertex can have $\max\sigma + \max\rho + 2$ states. (σ, ρ are finite!)
$\rightsquigarrow$ For a set of k vertices, $(\max\sigma + \max\rho + 2)^k$ combinations must be to considered

Assume (σ, ρ) is m-structured (e.g., $m = 2$):

- The selection status of specific vertices determines the number of neighbors for other vertices (think of it as some parity-like argument).
- Selection state and number of neighbors are "orthogonal" properties

A Better Bound for the Number of States

Recall: Each vertex can have $\max \sigma + \max \rho + 2$ states. (σ, ρ are finite!)
$\rightsquigarrow$ For a set of k vertices, $(\max \sigma + \max \rho + 2)^k$ combinations must be to considered

Assume (σ, ρ) is m-structured (e.g., $m = 2$):

- The selection status of specific vertices determines the number of neighbors for other vertices (think of it as some parity-like argument).
- Selection state and number of neighbors are "orthogonal" properties

$\rightsquigarrow$ Many combinations are ruled out

A Better Bound for the Number of States

Recall: Each vertex can have $\max\sigma + \max\rho + 2$ states. (σ, ρ are finite!)
$\rightsquigarrow$ For a set of k vertices, $(\max\sigma + \max\rho + 2)^k$ combinations must be to considered

Assume (σ, ρ) is m-structured (e.g., $\mathrm{m} = 2$):

- The selection status of specific vertices determines the number of neighbors for other vertices (think of it as some parity-like argument).
- Selection state and number of neighbors are "orthogonal" properties

$\rightsquigarrow$ Many combinations are ruled out

Key Lemma (Upper Bounds)

For m-structured (σ, ρ), the number of partial solutions for a separator of size k is

- $\approx (\max(\sigma \cup \rho) + 2)^k$ for $\mathrm{m} = 2$ and $\max\sigma = \max\rho$ even,
- $\approx (\max(\sigma \cup \rho) + 1)^k$ otherwise.

Decision Version

- Algorithm from counting version transfers naturally, but more cases can be solved trivially (e.g., if $0 \in \rho$)
- Lower bounds can be adjusted for finite σ and ρ

Decision Version

- Algorithm from counting version transfers naturally, but more cases can be solved trivially (e.g., if $0 \in \rho$)
- Lower bounds can be adjusted for finite σ and ρ

For cofinite σ or ρ:

- Use technique of representative sets to store fewer states

Decision Version

- Algorithm from counting version transfers naturally, but more cases can be solved trivially (e.g., if $0 \in \rho$)
- Lower bounds can be adjusted for finite σ and ρ

For cofinite σ or ρ:

- Use technique of representative sets to store fewer states
- Number of partial solutions stored $\approx$ number of *missing* degrees $\rightsquigarrow$ better algorithm

Decision Version

- Algorithm from counting version transfers naturally, but more cases can be solved trivially (e.g., if $0 \in \rho$)
- Lower bounds can be adjusted for finite σ and ρ

For cofinite σ or ρ:

- Use technique of representative sets to store fewer states
- Number of partial solutions stored $\approx$ number of *missing* degrees $\rightsquigarrow$ better algorithm

Definition: $\text{cost}(\tau) := \max(\tau)$ if τ is finite and $\text{cost}(\tau) := |\mathbb{N} \setminus \tau|$ if τ is cofinite.

Decision Version

- Algorithm from counting version transfers naturally, but more cases can be solved trivially (e.g., if $0 \in \rho$)
- Lower bounds can be adjusted for finite σ and ρ

For cofinite σ or ρ:

- Use technique of representative sets to store fewer states
- Number of partial solutions stored $\approx$ number of *missing* degrees $\rightsquigarrow$ better algorithm

Definition: $\mathrm{cost}(\tau) := \max(\tau)$ if τ is finite and $\mathrm{cost}(\tau) := |\mathbb{N} \setminus \tau|$ if τ is cofinite.

Theorem

(σ, ρ)-Dominating Set can be solved in time $(\max\{\mathrm{cost}(\sigma), \mathrm{cost}(\rho)\} + 1)^{(\omega+2)\mathrm{tw}} \cdot n^{\mathcal{O}(1)}$ for (co)finite σ, ρ, given a tree decomposition of width tw with ω as matrix multiplication exponent.

Decision Version

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

- Algorithm from counting version transfers naturally, but more cases can be solved trivially (e.g., if $0 \in \rho$)
- Lower bounds can be adjusted for finite σ and ρ

For cofinite σ or ρ:

- Use technique of representative sets to store fewer states
- Number of partial solutions stored $\approx$ number of *missing* degrees $\rightsquigarrow$ better algorithm

Definition: $\mathrm{cost}(\tau) := \max(\tau)$ if τ is finite and $\mathrm{cost}(\tau) := |\mathbb{N} \setminus \tau|$ if τ is cofinite.

Theorem

(σ, ρ)-Dominating Set can be solved in time $(\max\{\mathrm{cost}(\sigma), \mathrm{cost}(\rho)\} + 1)^{(\omega+2)\mathrm{tw}} \cdot n^{\mathcal{O}(1)}$ for (co)finite σ, ρ, given a tree decomposition of width tw with ω as matrix multiplication exponent.

Example for $\sigma = \{0\}$, $\rho = \mathbb{N} \setminus \{1000\}$:
The previous 1002^{tw} algorithm is improved to get a $2^{(\omega+2)\mathrm{tw}} < 21^{\mathrm{tw}}$ algorithm

Summary

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not m $\geq$ 3-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

Summary

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{top} + \rho_{top} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 2$ if $\sigma_{top} = \rho_{top}$ is even and (σ, ρ) is 2-structured (but not $m \geq 3$-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 1$ otherwise.

- Optimal $(c_{\sigma,\rho})^{tw} \cdot \mathrm{poly}(|G|)$ algo for counting version, unless #SETH fails.

Summary

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{top} + \rho_{top} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 2$ if $\sigma_{top} = \rho_{top}$ is even and (σ, ρ) is 2-structured (but not m $\geq$ 3-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{top}, \rho_{top}\} + 1$ otherwise.

- Optimal $(c_{\sigma,\rho})^{tw} \cdot \text{poly}(|G|)$ algo for counting version, unless #SETH fails.
- Optimal $(c_{\sigma,\rho})^{tw} \cdot \text{poly}(|G|)$ algo for decision version with *finite* σ and ρ, unless SETH fails.

Summary

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not m $\geq$ 3-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

- Optimal $(c_{\sigma,\rho})^{\text{tw}} \cdot \text{poly}(|G|)$ algo for counting version, unless #SETH fails.
- Optimal $(c_{\sigma,\rho})^{\text{tw}} \cdot \text{poly}(|G|)$ algo for decision version with *finite* σ and ρ, unless SETH fails.
- Faster algorithm for decision version if (at least) one of σ and ρ is *cofinite*.

Summary

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not $m \geq 3$-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

- Optimal $(c_{\sigma,\rho})^{\text{tw}} \cdot \text{poly}(|G|)$ algo for counting version, unless #SETH fails.
- Optimal $(c_{\sigma,\rho})^{\text{tw}} \cdot \text{poly}(|G|)$ algo for decision version with *finite* σ and ρ, unless SETH fails.
- Faster algorithm for decision version if (at least) one of σ and ρ is *cofinite*.

Full paper: `arxiv.org/abs/2211.04278`

Summary (formalized)

Definition

For finite or cofinite $\sigma, \rho \subseteq \mathbb{N}$, we set

- $c_{\sigma,\rho} := \sigma_{\text{top}} + \rho_{\text{top}} + 2$ if (σ, ρ) is not m-structured,
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 2$ if $\sigma_{\text{top}} = \rho_{\text{top}}$ is even and (σ, ρ) is 2-structured (but not $m \geq 3$-structured), and
- $c_{\sigma,\rho} := \max\{\sigma_{\text{top}}, \rho_{\text{top}}\} + 1$ otherwise.

Theorem (Upper Bounds)

$\#(\sigma, \rho)$-Dominating Set *can be solved in time* $(c_{\sigma,\rho})^{\text{tw}(G)} \cdot \text{poly}(|G|)$

if σ, ρ *are finite or cofinite and a tree decomposition of width* $\text{tw}(G)$ *is given.*

Theorem (Lower Bounds)

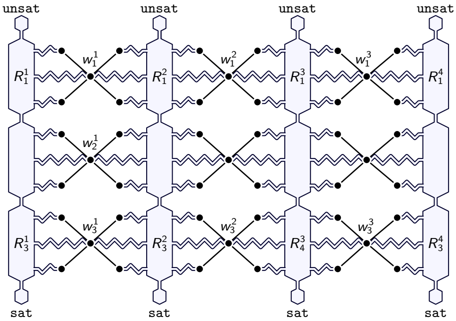
Unless #SETH fails, $\#(\sigma, \rho)$-Dominating Set *has no* $(c_{\sigma,\rho} - \varepsilon)^{\text{tw}(G)} \cdot \text{poly}(|G|)$ *algo*

for non-trivial, finite or cofinite σ, ρ*, even if a tree decomposition of width* $\text{tw}(G)$ *is given.*

Lower Bound

Use general idea from [CM16]:

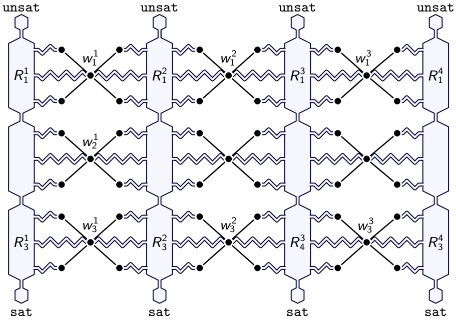
1. Reduce #SAT to #(σ, ρ)-Dominating Set ***with constraints***
   - Constraints restrict the selection of vertices to predefined combinations



Lower Bound

Use general idea from [CM16]:

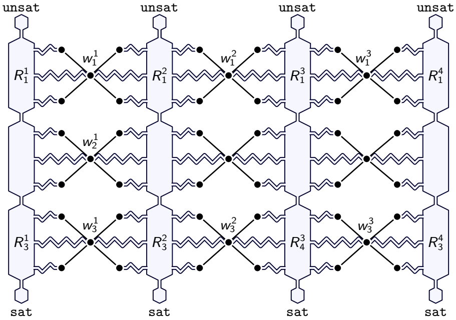
1. Reduce #SAT to #(σ, ρ)-Dominating Set *with constraints*
   - Constraints restrict the selection of vertices to predefined combinations
   - This provides the power to check if "clauses are satisfied"



Lower Bound

Use general idea from [CM16]:

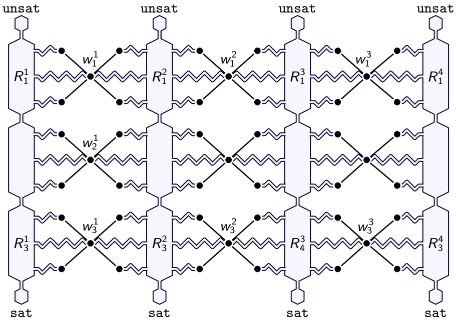
1. Reduce #SAT to #(σ, ρ)-Dominating Set *with constraints*
   - Constraints restrict the selection of vertices to predefined combinations
   - This provides the power to check if “clauses are satisfied”
2. Implement the constraints



Lower Bound

Use general idea from [CM16]:

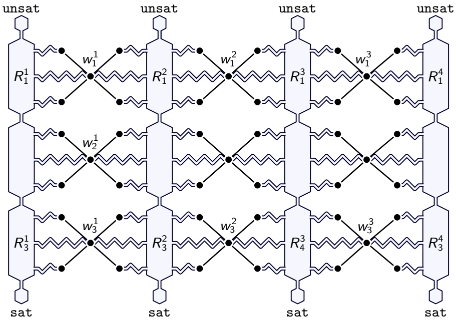
1. Reduce #SAT to #(σ, ρ)-DOMINATING SET *with constraints*
   - Constraints restrict the selection of vertices to predefined combinations
   - This provides the power to check if "clauses are satisfied"
2. Implement the constraints
   - Solutions that are easy to *find*, have to be counted, too.



Lower Bound

Use general idea from [CM16]:

1. Reduce #SAT to #(σ, ρ)-Dominating Set *with constraints*
   - Constraints restrict the selection of vertices to predefined combinations
   - This provides the power to check if "clauses are satisfied"
2. Implement the constraints
   - Solutions that are easy to *find*, have to be counted, too.
   - Use counting complexity techniques and carefully designed gadgets to overcome these issues



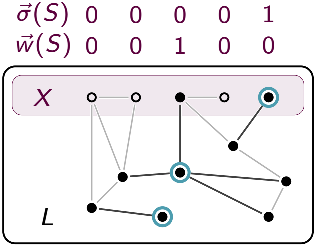
Decomposing Types Helps

Consider Perfect Code ($\sigma = \{0\}$, $\rho = \{1\}$, m-structured for all $m \geq 2$)

Decomposing Types Helps

Consider PERFECT CODE ($\sigma = \{0\}$, $\rho = \{1\}$, m-structured for all $m \geq 2$)

- For a separator X and a partial solution S, we define σ vector $\overrightarrow{\sigma}(S) \in \{0,1\}^{|X|}$ and weight vector $\overrightarrow{w}(S) \in \{0,1\}^{|X|}$

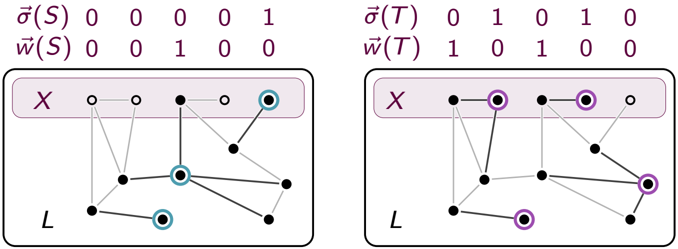


Decomposing Types Helps

Consider Perfect Code ($\sigma = \{0\}$, $\rho = \{1\}$, m-structured for all $m \geq 2$)

- For a separator X and a partial solution S, we define σ vector $\vec{\sigma}(S) \in \{0,1\}^{|X|}$ and weight vector $\vec{w}(S) \in \{0,1\}^{|X|}$

Consider two partial solutions S, T for X:



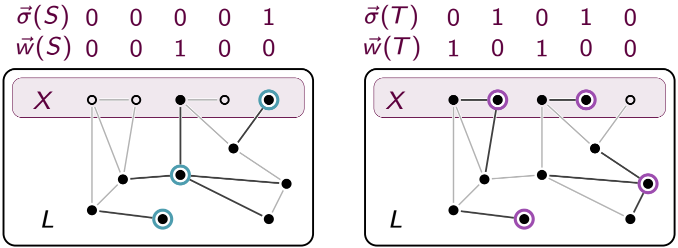
Decomposing Types Helps

Consider Perfect Code ($\sigma = \{0\}$, $\rho = \{1\}$, m-structured for all $m \geq 2$)

- For a separator X and a partial solution S, we define σ vector $\vec{\sigma}(S) \in \{0,1\}^{|X|}$ and weight vector $\vec{w}(S) \in \{0,1\}^{|X|}$

Consider two partial solutions S, T for X:

- Observe: $\vec{\sigma}(S) \cdot \vec{w}(T) \equiv_m \vec{\sigma}(T) \cdot \vec{w}(S)$



Decomposing Types Helps

Consider Perfect Code ($\sigma = \{0\}$, $\rho = \{1\}$, m-structured for all $m \geq 2$)

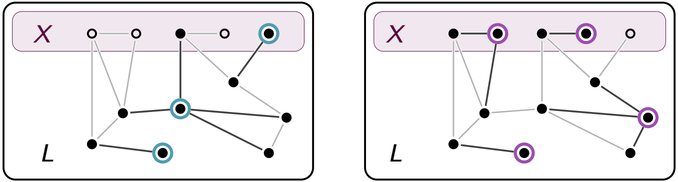
- For a separator X and a partial solution S, we define σ vector $\overrightarrow{\sigma}(S) \in \{0,1\}^{|X|}$ and weight vector $\overrightarrow{w}(S) \in \{0,1\}^{|X|}$

Consider two partial solutions S, T for X:

- Observe: $\overrightarrow{\sigma}(S) \cdot \overrightarrow{w}(T) \equiv_m \overrightarrow{\sigma}(T) \cdot \overrightarrow{w}(S)$ $\rightsquigarrow$ σ-vectors are “orthogonal” to weight-vectors
- Not too many solutions satisfy this “orthogonality” at the same time

$\vec{\sigma}(S)$	0	0	0	0	1
$\vec{w}(S)$	0	0	1	0	0

$\vec{\sigma}(T)$	0	1	0	1	0
$\vec{w}(T)$	1	0	1	0	0



Decomposing Types Helps

Consider Perfect Code ($\sigma = \{0\}$, $\rho = \{1\}$, m-structured for all $m \geq 2$)

- For a separator X and a partial solution S, we define σ vector $\vec{\sigma}(S) \in \{0,1\}^{|X|}$ and weight vector $\vec{w}(S) \in \{0,1\}^{|X|}$

Consider two partial solutions S, T for X:

- Observe: $\vec{\sigma}(S) \cdot \vec{w}(T) \equiv_m \vec{\sigma}(T) \cdot \vec{w}(S)$ $\rightsquigarrow$ σ-vectors are "orthogonal" to weight-vectors
- Not too many solutions satisfy this "orthogonality" at the same time

Key Lemma (Upper Bounds)

For m-structured (σ, ρ), the number of partial solutions is

- $\mathcal{O}((\max\{\sigma_{\mathrm{top}}, \rho_{\mathrm{top}}\} + 2)^{\mathrm{tw}})$ for $m = 2$ and $\sigma_{\mathrm{top}} = \rho_{\mathrm{top}}$ even,
- $\mathcal{O}((\max\{\sigma_{\mathrm{top}}, \rho_{\mathrm{top}}\} + 1)^{\mathrm{tw}})$ otherwise.

Lower Bounds

Use general framework from [CM16]:

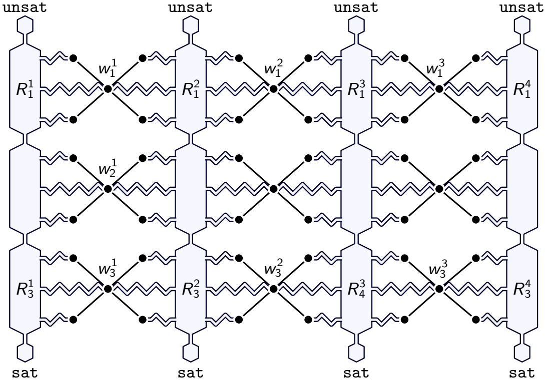
1. Reduce #SAT to #(σ, ρ)-Dominating Set *with constraints*
2. Implement constraints using carefully crafted gadgets

Lower Bounds

Use general framework from [CM16]:

1. **Reduce #SAT to #(σ, ρ)-Dominating Set *with constraints***
2. Implement constraints using carefully crafted gadgets

- group $\approx \log(\sigma_{\text{top}} + \rho_{\text{top}} + 2)$ variables; one row for each group
- one column for each clause

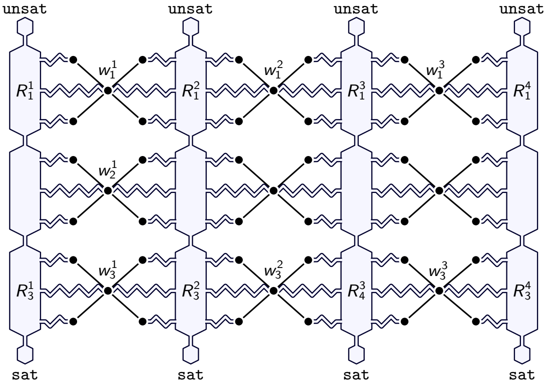


Lower Bounds

Use general framework from [CM16]:

1. **Reduce #SAT to #(σ, ρ)-Dominating Set *with constraints***
2. Implement constraints using carefully crafted gadgets

- group $\approx \log(\sigma_{\text{top}} + \rho_{\text{top}} + 2)$ variables; one row for each group
- one column for each clause
- states of information vertices encode the assignment for each group

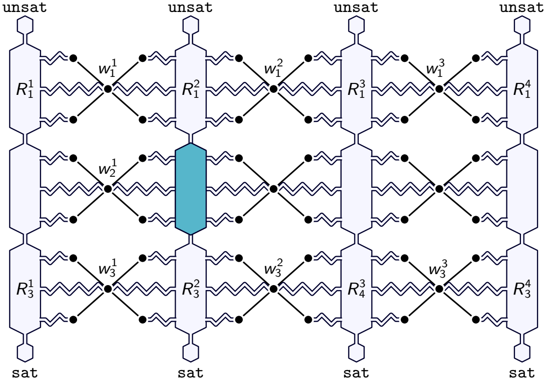


Lower Bounds

Use general framework from [CM16]:

1. **Reduce #SAT to #(σ, ρ)-Dominating Set *with constraints***
2. Implement constraints using carefully crafted gadgets

- group $\approx \log(\sigma_{\text{top}} + \rho_{\text{top}} + 2)$ variables; one row for each group
- one column for each clause
- states of information vertices encode the assignment for each group
- relation R_i^j checks if assignment to ith group satisfies jth clause

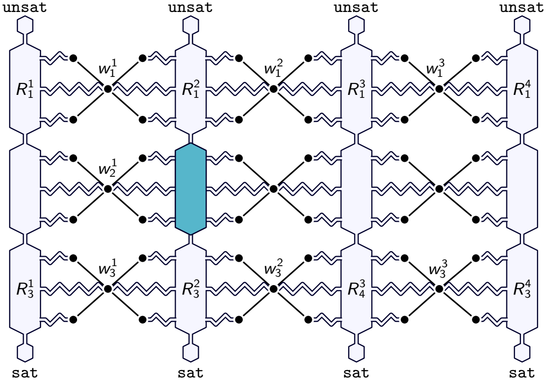


Lower Bounds

Use general framework from [CM16]:

1. Reduce #SAT to #(σ, ρ)-DOMINATING SET *with constraints*
2. **Implement constraints using carefully crafted gadgets**

- group $\approx \log(\sigma_{top} + \rho_{top} + 2)$ variables; one row for each group
- one column for each clause
- states of information vertices encode the assignment for each group
- relation R_i^j checks if assignment to ith group satisfies jth clause
- relations are realized using *Hamming weight* $= 1$ and *equality*

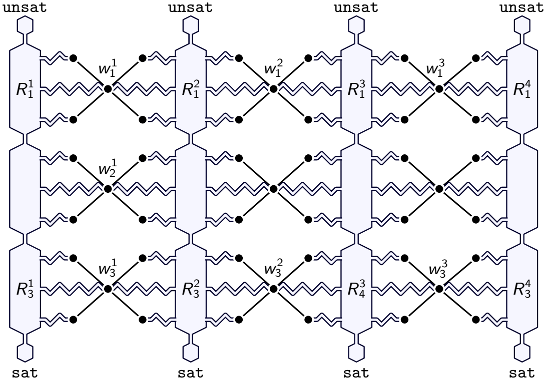


Lower Bounds

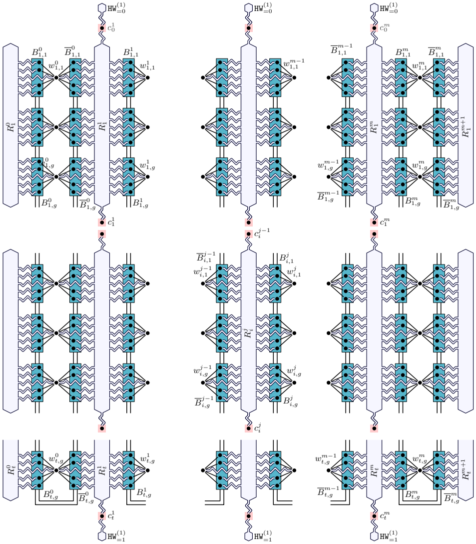
Use general framework from [CM16]:

1. Reduce #SAT to #(σ,ρ)-DOMINATING SET *with constraints*
2. Implement constraints using carefully crafted gadgets

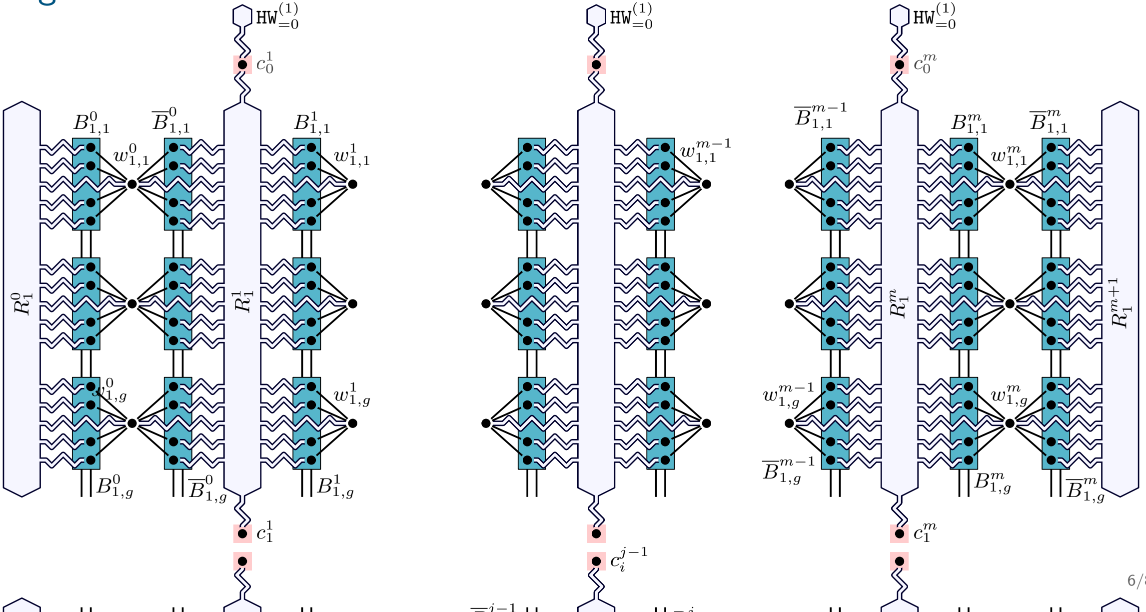
- group $\approx \log(\sigma_{\text{top}} + \rho_{\text{top}} + 2)$ variables; one row for each group
- one column for each clause
- states of information vertices encode the assignment for each group
- relation R_i^j checks if assignment to ith group satisfies jth clause
- relations are realized using *Hamming weight* $= 1$ and *equality*



High Level Construction

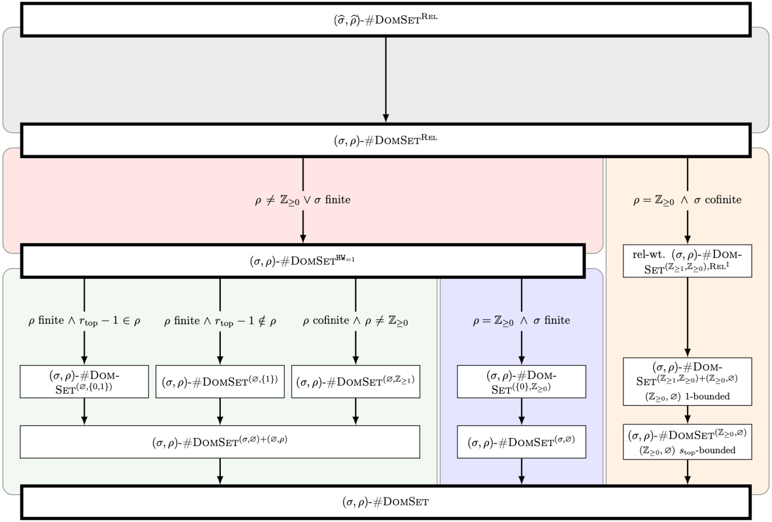


High Level Construction



Removing the Relations in the Counting Version (simplified)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY



Removing the Relations in the Counting Version