UNIVERSITÄT ZU LÜBECK

MaxSAT: Parameterized, Parallel, Absolute

Max Bannach Pamela Fleischmann

Malte Skambath Till Tantau

August 1, 2022

IM FOCUS DAS LEBEN

Very Classical MaxSAT

Problem: p_k-MAX-SAT

Instance	A propositional formula φ in conjunctive normal form and an integer k.
Parameter	k
Question	Is there an assignment $\beta\colon \operatorname{vars}(\varphi) \to \{0, 1\}$ that satisfies at least k clauses?

Very Classical MaxSAT

Problem: p_k-MAX-SAT

Instance	A propositional formula φ in conjunctive normal form and an integer k.
Parameter	k
Question	Is there an assignment $\beta\colon \operatorname{vars}(\varphi) \to \{0,1\}$ that satisfies at least k clauses?

Example

$$\varphi \equiv (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (x_3)$$

Very Classical MaxSAT

Problem: p_k-MAX-SAT

Instance	A propositional formula φ in conjunctive normal form and an integer k.
Parameter	k
Question	Is there an assignment $\beta\colon \text{vars}(\varphi) \to \{0,1\}$ that satisfies at least k clauses?

Example

$$\varphi \equiv (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor \neg x_2) \land (\neg x_1 \lor \neg x_2) \land (x_1 \lor x_2 \lor \neg x_3) \land (x_3)$$

$$\beta_1 = [x_1 = 0, x_2 = 0, x_3 = 0]$$

Very Classical MaxSAT

Problem: p_k-MAX-SAT

Instance	A propositional formula φ in conjunctive normal form and an integer k.
Parameter	k
Question	Is there an assignment $\beta\colon \text{vars}(\varphi) \to \{0, 1\}$ that satisfies at least k clauses?

Example

$$\begin{aligned}\varphi \equiv\ & (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor \neg x_2) \land (\neg x_1 \lor \neg x_2) \\ & \land (x_1 \lor x_2 \lor \neg x_3) \land (x_3)\end{aligned}$$

$$\beta_1 = [x_1 = 0, x_2 = 0, x_3 = 0]$$

$$\beta_2 = [x_1 = 0, x_2 = 0, x_3 = 1]$$

Very Classical MaxSAT

Problem: p_k-MAX-SAT

Instance A propositional formula φ in conjunctive normal form and an integer k.

Parameter k

Question Is there an assignment $\beta\colon \operatorname{vars}(\varphi) \to \{0,1\}$ that satisfies at least k clauses?

Example

$$\varphi \equiv (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor x_2) \land (x_1 \lor \neg x_2) \land (\neg x_1 \lor \neg x_2) \\ \land (x_1 \lor x_2 \lor \neg x_3) \land (x_3)$$

$$\beta_1 = [x_1 = 0, x_2 = 0, x_3 = 0]$$

$$\beta_2 = [x_1 = 0, x_2 = 0, x_3 = 1]$$

For now without any extras (no weights, no bound on the clauses size, …).

Overview of the Results

Problem	para-	P
p$_k$-MAX-SAT		✓

A “better” Parameter

- This result is medium interesting.
  - A “good” parameter is reasonable small on reasonable many instances.

A “better” Parameter

- This result is medium interesting.
  - A “good” parameter is reasonable small on reasonable many instances.
- Any formula has an assignment that satisfies at least half of the clauses.
  - Consider an arbitrary assignment and its complement.

A “better” Parameter

- This result is medium interesting.
    - A “good” parameter is reasonable small on reasonable many instances.
- Any formula has an assignment that satisfies at least half of the clauses.
    - Consider an arbitrary assignment and its complement.

Problem: p_g-MAX-SAT-ABOVE-HALF

Instance	A propositional formula φ in conjunctive normal form and an integer k.
Parameter	$g = k - \lceil \frac{m}{2} \rceil$
Question	Is there an assignment $\beta\colon \text{vars}(\varphi) \to \{0, 1\}$ that satisfies at least k clauses?

Overview of the Results

Problem	para- P
p_k-MAX-SAT	✓

Overview of the Results

Problem	para-	P
p_k-MAX-SAT		✓
p_g-MAX-SAT-ABOVE-HALF		✓

Table of Contents

Table of Contents

Parallel Parameterized Complexity

Definition: para-AC^0

Parameterized problems solvable by a uniform family of circuits with:

Base	$\{\vee, \wedge, \neg\}$
Fan-In	unbounded
Depth	$O(1)$
Size	$f(k) \cdot \mathrm{poly}(n)$

A Simple Parallel Algorithm

Theorem

p_k-MAX-SAT $\in$ para-AC^0

Proof Sketch:

- The probability that a random assignment satisfies at least k clauses depends only on k.
  - (if such an assignment exists)
- It is therefore sufficient to test $f(k)$ random assignments in parallel.
- This process can be derandomized using universal hash functions.
  - (color coding)

☐

A Simple Parallel Algorithm

Theorem

p_k-MAX-SAT $\in$ para-AC^0

Proof Sketch:

- The probability that a random assignment satisfies at least k clauses depends only on k.
  - (if such an assignment exists)
- It is therefore sufficient to test $f(k)$ random assignments in parallel.
- This process can be derandomized using universal hash functions.
  - (color coding)

☐

A Simple Parallel Algorithm

Theorem

p_k-MAX-SAT $\in$ para-AC^0

Proof Sketch:

- The probability that a random assignment satisfies at least k clauses depends only on k.
  - (if such an assignment exists)
- It is therefore sufficient to test $f(k)$ random assignments in parallel.
- This process can be derandomized using universal hash functions.
  - (color coding)

□

A Simple Parallel Algorithm

Theorem

p_k-MAX-SAT $\in$ para-AC^0

Proof Sketch:

- The probability that a random assignment satisfies at least k clauses depends only on k.
  - (if such an assignment exists)

- It is therefore sufficient to test $f(k)$ random assignments in parallel.

- This process can be derandomized using universal hash functions.
  - (color coding)

$\square$

Overview of the Results

	para-	
Problem		P
p_k-MAX-SAT		✓
p_g-MAX-SAT-ABOVE-HALF		✓

Overview of the Results

Problem	para-AC^0	P
p_k-MAX-SAT	✓	✓
p_g-MAX-SAT-ABOVE-HALF		✓

Parallel Parameterized Complexity

Definition: para-TC^0

Parameterized problems solvable by a uniform family of circuits with:

Base	$\{\vee, \wedge, \neg, \text{maj}\}$
Fan-In	unbounded
Depth	$O(1)$
Size	$f(k) \cdot \text{poly}(n)$

para-AC^0 ➔ para-TC^0

Parallel Data Reduction

Theorem

p_g-MAXSAT-ABOVE-HALF is para-TC^0-complete.

Proof Sketch.

Use the following reduction rules due to Mahajan and Raman:

Parallel Data Reduction

Theorem

p_g-MAXSAT-ABOVE-HALF is para-TC^0-complete.

Proof Sketch.

Use the following reduction rules due to Mahajan and Raman:

Rule (Empty Clauses)

If there are empty clauses, remove them.

Parallel Data Reduction

Theorem

p_g-MAXSAT-ABOVE-HALF is para-TC^0-complete.

Proof Sketch.

Use the following reduction rules due to Mahajan and Raman:

Rule (Empty Clauses)

If there are empty clauses, remove them.

Rule (Unit Pair)

If there are two unit clauses (x) *and* $(\neg x)$*, remove both.*

Parallel Data Reduction

Theorem

p_g-MAXSAT-ABOVE-HALF is para-TC^0-complete.

Proof Sketch.

Use the following reduction rules due to Mahajan and Raman:

Rule (Empty Clauses)

If there are empty clauses, remove them.

Rule (Unit Pair)

If there are two unit clauses (x) *and* $(\neg x)$*, remove both.*

Rule (Trivial Decision)

Reduce to a trivial yes-instance if there are at least $\lceil m/2 \rceil + k$ *unit clauses or at least* $4k + 4$ *non-unit clauses.*

Parallel Data Reduction

Theorem

p_g-MAXSAT-ABOVE-HALF is para-TC^0-complete.

Proof Sketch.

Use the following reduction rules due to Mahajan and Raman:

Rule (Empty Clauses)

If there are empty clauses, remove them.

Rule (Unit Pair)

If there are two unit clauses (x) *and* $(\neg x)$*, remove both.*

Rule (Trivial Decision)

Reduce to a trivial yes-instance if there are at least $\lceil m/2 \rceil + k$ *unit clauses or at least* $4k + 4$ *non-unit clauses.*

If we have not decided we have $\lfloor m/2 \rfloor \leq 5k + 2$ and, thus, a p_k-MAX-SAT instance with parameter $6k + 3$.

Parallel Data Reduction

Theorem

p_g-MAXSAT-ABOVE-HALF is para-TC^0-complete.

Proof Sketch (continued).

For hardness, reduce from p_0-MAJORITY.

- Does a binary string contain at least $n/2$ ones?
- Trivial parameter (constant 0).

(For circuit experts: This is a para-AC^0-truth-table reduction.) □

Overview of the Results

Problem	para- AC^0	P
p_k-MAX-SAT	✓	✓
p_g-MAX-SAT-ABOVE-HALF		✓

Overview of the Results

Problem	para- AC^0	TC^0	P
p_k-MAX-SAT	✓		✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓	✓

Parallel Parameterized Complexity

Definition: para-NC^i

Parameterized problems solvable by a uniform family of circuits with:

Base	$\{\vee, \wedge, \neg\}$
Fan-In	bounded
Depth	$f(k) + O(\log^i n)$
Size	$f(k) \cdot \mathrm{poly}(n)$

para-AC^0 → para-TC^0 → para-NC^1

Parallel Parameterized Complexity

Definition: para-AC^i

Parameterized problems solvable by a uniform family of circuits with:

Base	$\{\vee, \wedge, \neg\}$
Fan-In	unbounded
Depth	$f(k) + O(\log^i n)$
Size	$f(k) \cdot \text{poly}(n)$

$\text{para-AC}^0 \rightarrow \text{para-TC}^0 \rightarrow \text{para-NC}^1 \longrightarrow \text{para-L} \longrightarrow \text{para-NL} \rightarrow \text{para-AC}^1 \rightarrow \text{para-AC}^2$

Parallel Kernelization

Theorem

A decidable problem is in para-P *if, and only if, it has a kernelization in* P.

So on the previous slide, both problems have a kernelization.

Parallel Kernelization

Theorem

A decidable problem is in para-P *if, and only if, it has a kernelization in* P.

So on the previous slide, both problems have a kernelization.

Theorem

A decidable problem is in para-AC^i *if, and only if, it has a kernelization in* AC^i.

Previous results can be seen as parallel kernelizations.

Parallel Kernelization for MaxSAT

- p_k-MAX-SAT has a kernelization …
  - … computable in AC^0
  - … computable constant parallel time with polynomial work
  - … describable in FO-logic

- the same is *not* true for p_k-MAX-SAT-ABOVE-HALF

Another Example: MaxDNF

Problem: $\mathrm{p}_{k,d}$-MAX-DNF

Instance	A propositional formula φ in *disjunctive* normal form with *terms of size at most d* and an integer k.
Parameter	k, d
Question	Is there an assignment $\beta\colon \operatorname{vars}(\varphi) \to \{0,1\}$ that satisfies at least k terms?

Another Example: MaxDNF

Problem: $\text{p}_{k,d}$-MAX-DNF

Instance	A propositional formula φ in *disjunctive* normal form with *terms of size at most d* and an integer k.
Parameter	k, d
Question	Is there an assignment $\beta\colon \text{vars}(\varphi) \to \{0, 1\}$ that satisfies at least k terms?

Theorem

$\text{p}_{k,d}$-MAX-DNF $\in$ para-AC^0

Proof.

Anyone has an idea? $\square$

Overview of the Results

Problem	para-AC^0	TC^0	P
p_k-MAX-SAT	✓		✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓	✓

Overview of the Results

Problem	para- AC^0	TC^0	P
p_k-MAX-SAT	✓		✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓	✓
$p_{k,d}$-MAX-DNF	✓		✓

Overview of the Results

Problem	para-AC^0	TC^0	P
Parameterized by number of clauses to be satisfied (solution size):			
p_k-MAX-SAT	✓		✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓	✓
p_k-MAX-NAE-SAT	✓		✓
$\mathrm{p}_{k,d}$-MAX-DNF	✓		✓

Another Example: AlmostDNF

Problem: p_k-ALMOST-DNF

Instance	A propositional formula φ in disjunctive normal form and an integer k.
Parameter	k
Question	Is there an assignment $\beta\colon \operatorname{vars}(\varphi) \to \{0, 1\}$ that satisfies at least $m - k$ terms?

Another Example: AlmostDNF

Problem: p_k-ALMOST-DNF

Instance	A propositional formula φ in disjunctive normal form and an integer k.
Parameter	k
Question	Is there an assignment $\beta \colon \operatorname{vars}(\varphi) \to \{0, 1\}$ that satisfies at least $m - k$ terms?

- From a parameterized perspective this is a “tighter” parameter.
- For CNFs the problem contains SAT (for $k = 0$).

Another Example: AlmostDNF

Theorem

p_k-ALMOST-DNF $\in$ para-AC^0

Sketch of Proof.

- Replace each term $(\ell_1 \land \ldots \land \ell_d)$ with the clause $(\neg\ell_1 \lor \ldots \lor \neg\ell_d)$.
- A clause is satisfied iff the term is not satisfied.
  - Hence, we have to find an assignment that satisfies *at most k* clauses.
  - This is p_k-MIN-SAT.
- p_k-MIN-SAT reduced to p_k-VERTEX-COVER.
  - Construct a graph that contains a vertex for each clause.
  - Connect two vertices if the corresponding clauses contain complementary literals.

☐

Another Example: AlmostDNF

Theorem

p_k-ALMOST-DNF $\in$ para-AC^0

Sketch of Proof.

- Replace each term $(\ell_1 \wedge \ldots \wedge \ell_d)$ with the clause $(\neg\ell_1 \vee \ldots \vee \neg\ell_d)$.
- A clause is satisfied iff the term is not satisfied.
  - Hence, we have to find an assignment that satisfies *at most k* clauses.
  - This is p_k-MIN-SAT.
- p_k-MIN-SAT reduced to p_k-VERTEX-COVER.
  - Construct a graph that contains a vertex for each clause.
  - Connect two vertices if the corresponding clauses contain complementary literals.

☐

Another Example: AlmostDNF

Theorem

p_k-ALMOST-DNF $\in$ para-AC^0

Sketch of Proof.

- Replace each term $(\ell_1 \wedge \ldots \wedge \ell_d)$ with the clause $(\neg\ell_1 \vee \ldots \vee \neg\ell_d)$.
- A clause is satisfied iff the term is not satisfied.
  - Hence, we have to find an assignment that satisfies *at most k* clauses.
  - This is p_k-MIN-SAT.
- p_k-MIN-SAT reduced to p_k-VERTEX-COVER.
  - Construct a graph that contains a vertex for each clause.
  - Connect two vertices if the corresponding clauses contain complementary literals.

☐

Another Example: AlmostDNF

Theorem

p$_k$-ALMOST-DNF $\in$ para-AC^0

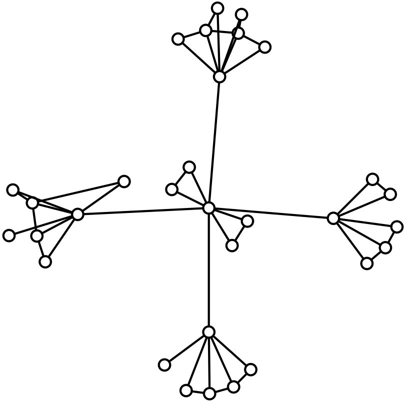
Sketch of Proof.

- Replace each term $(\ell_1 \wedge \ldots \wedge \ell_d)$ with the clause $(\neg\ell_1 \vee \ldots \vee \neg\ell_d)$.
- A clause is satisfied iff the term is not satisfied.
  - Hence, we have to find an assignment that satisfies *at most k* clauses.
  - This is p$_k$-MIN-SAT.
- p$_k$-MIN-SAT reduced to p$_k$-VERTEX-COVER.
  - Construct a graph that contains a vertex for each clause.
  - Connect two vertices if the corresponding clauses contain complementary literals.

☐

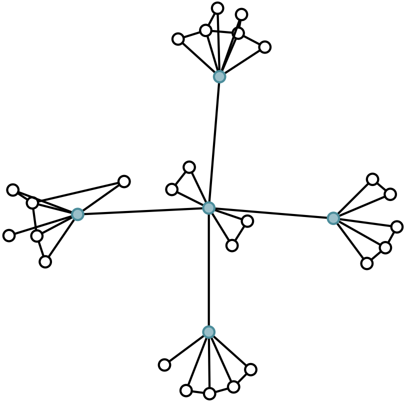
A Parallel Kernel for Vertex Cover

- check if $k \leq \log n$



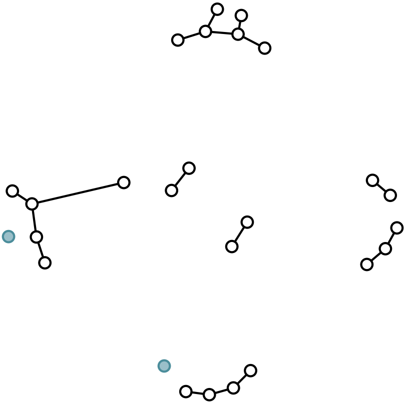
A Parallel Kernel for Vertex Cover

- check if $k \leq \log n$
- Delete vertices with $|N(v)| > k$.



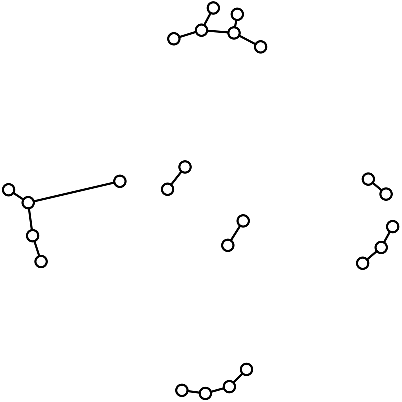
A Parallel Kernel for Vertex Cover

- check if $k \leq \log n$
- Delete vertices with $|N(v)| > k$.
- Delete vertices with $|N(v)| = 0$.



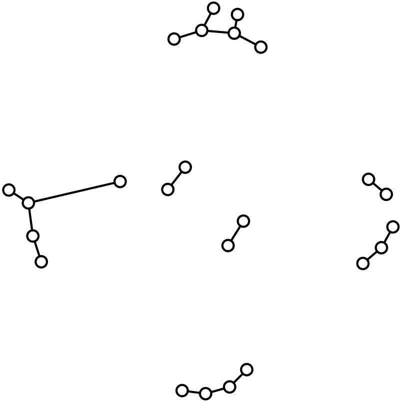
A Parallel Kernel for Vertex Cover

- check if $k \leq \log n$
- Delete vertices with $|N(v)| > k$.
- Delete vertices with $|N(v)| = 0$.
- At most $k^2 + k$ vertices can remain.



A Parallel Kernel for Vertex Cover

- check if $k \leq \log n$
- Delete vertices with $|N(v)| > k$.
- Delete vertices with $|N(v)| = 0$.
- At most $k^2 + k$ vertices can remain.



Lemma
A p_k-VERTEX-COVER *kernel can be computed by* AC^0*-circuits.*

Overview of the Results

Problem	para- AC^0	TC^0	P
Parameterized by number of clauses to be satisfied (solution size):			
p_k-MAX-SAT	✓		✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓	✓
p_k-MAX-NAE-SAT	✓		✓
$p_{k,d}$-MAX-DNF	✓		✓

Overview of the Results

Problem	para-AC^0	TC^0	P
Parameterized by number of clauses to be satisfied (solution size):			
p_k-MAX-SAT	✓		✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓	✓
p_k-MAX-NAE-SAT	✓		✓
$\mathrm{p}_{k,d}$-MAX-DNF	✓		✓
Parameterized by number of clauses left unsatisfied (dual parameter):			
p_k-ALMOST-DNF	✓		✓

Overview of the Results

	para-								
Problem	AC^0	TC^0	$TC^{0\uparrow}$	L	$L^{\uparrow}$	$NL^{\uparrow}$	$TC^{1\uparrow}$	$AC^{2\uparrow}$	P
Parameterized by number of clauses to be satisfied (solution size):									
p_k-MAX-SAT	✓								✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓							✓
p_k-MAX-NAE-SAT	✓								✓
$p_{k,d}$-MAX-DNF	✓								✓
Parameterized by number of clauses left unsatisfied (dual parameter):									
p_k-ALMOST-2SAT					✗	✓			✓
p_k-ALMOST-NAE-2SAT					?	✓			✓
p_k-ALMOST-NAE-SAT(2)			✗	✓					✓
p_k-ALMOST-SAT(2)			✗	✓					✓
p_k-ALMOST-DNF	✓								✓

Overview of the Results

	para-								
Problem	AC^0	TC^0	$TC^{0\uparrow}$	L	$L^{\uparrow}$	$NL^{\uparrow}$	$TC^{1\uparrow}$	$AC^{2\uparrow}$	P
Parameterized by number of clauses to be satisfied (solution size):									
p_k-MAX-SAT	✓								✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓							✓
p_k-MAX-NAE-SAT	✓								✓
$p_{k,d}$-MAX-DNF	✓								✓
Parameterized by number of clauses left unsatisfied (dual parameter):									
p_k-ALMOST-2SAT					✗	✓			✓
p_k-ALMOST-NAE-2SAT					?	✓			✓
p_k-ALMOST-NAE-SAT(2)			✗	✓					✓
p_k-ALMOST-SAT(2)			✗	✓					✓
p_k-ALMOST-DNF	✓								✓
Structural parameter:									
p_{vc}-PARTIAL-MAX-SAT	✗	✓							✓
p_{td}-PARTIAL-MAX-SAT	✗		✓						✓
p_{fvs}-PARTIAL-MAX-SAT			✗				✓		✓
p_{tw}-PARTIAL-MAX-SAT			✗					✓	✓

Table of Contents

Table of Contents

A Slightly Different Problem: MaxDNF with Weights

Problem: MAX-DNF

Instance A propositional formula φ in *disjunctive* normal form, a weight function $w\colon \text{terms}(\varphi) \to \mathbb{Z}$, and a target value $\alpha \in \mathbb{N}$.

Question Is there an assignment β such that the *total weight* of the satisfied terms is at least α?

A Slightly Different Problem: MaxDNF with Weights

Problem: MAX-DNF

Instance	A propositional formula φ in *disjunctive* normal form, a weight function $w\colon \operatorname{terms}(\varphi) \to \mathbb{Z}$, and a target value $\alpha \in \mathbb{N}$.
Question	Is there an assignment β such that the *total weight* of the satisfied terms is at least α?

Example

$$\varphi \equiv \underset{\substack{\downarrow \\ 2}}{(\neg x_1 \wedge x_2 \wedge x_3)} \vee \underset{\substack{\downarrow \\ -2}}{(\neg x_1 \wedge x_4)} \vee \underset{\substack{\downarrow \\ 1}}{(x_1)} \vee \underset{\substack{\downarrow \\ -1}}{(x_2 \wedge x_3)}$$

A Slightly Different Problem: MaxDNF with Weights

Problem: MAX-DNF

Instance A propositional formula φ in *disjunctive* normal form, a weight function $w\colon\ \mathrm{terms}(\varphi) \to \mathbb{Z}$, and a target value $\alpha \in \mathbb{N}$.

Question Is there an assignment β such that the *total weight* of the satisfied terms is at least α?

Example

$$\varphi \equiv \underset{\substack{\downarrow \\ 2}}{(\neg x_1 \wedge x_2 \wedge x_3)} \vee \underset{\substack{\downarrow \\ -2}}{(\neg x_1 \wedge x_4)} \vee \underset{\substack{\downarrow \\ 1}}{(x_1)} \vee \underset{\substack{\downarrow \\ -1}}{(x_2 \wedge x_3)}$$

$$\beta = [x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 0]$$

Variations of the Problem

Different variants of the problem:

- Formulas in conjunctive normal form (MAX-CNF).
- Allow/permit negative weights.
- Only positive literals / Monotone formulas (MAX-MONOTONE-DNF).

With Negative Weights the Problem is Hard

Theorem

*p_α-d-*MAX-MONOTONE-DNF *is* W[1]*-hard for $d \geq 2$.*

Proof Sketch

A graph $G = (V, E)$ has an independent set of size k iff

$$\varphi \equiv \bigvee_{v \in V} \underset{\substack{\downarrow \\ 1}}{(x_v)} \vee \bigvee_{\{v,w\} \in E} (x_v \wedge \underset{\substack{\downarrow \\ -1}}{x_w})$$

has an assignment of value $\alpha = k$.

Maximize the Absolute Value of the Sum

Problem: ABS-DNF

Instance A propositional formula φ in disjunctive normal form, a weight function $w\colon \text{terms}(\varphi) \to \mathbb{Z}$, and a target value $\alpha \in \mathbb{N}$.

Question Is there an assignment β such that the *absolute value* of the sum of the satisfied terms is at least α?

Maximize the Absolute Value of the Sum

Problem: ABS-DNF

Instance A propositional formula φ in disjunctive normal form, a weight function $w\colon \text{terms}(\varphi) \to \mathbb{Z}$, and a target value $\alpha \in \mathbb{N}$.

Question Is there an assignment β such that the *absolute value* of the sum of the satisfied terms is at least α?

Example

$$\varphi \equiv \underset{\substack{\downarrow\\ 2}}{(\neg x_1 \land x_2 \land x_3)} \lor \underset{\substack{\downarrow\\ -2}}{(\neg x_1 \land x_4)} \lor \underset{\substack{\downarrow\\ 1}}{(x_1)} \lor \underset{\substack{\downarrow\\ -1}}{(x_2 \land x_3)}$$

Maximize the Absolute Value of the Sum

Problem: ABS-DNF

Instance A propositional formula φ in disjunctive normal form, a weight function $w\colon \operatorname{terms}(\varphi) \to \mathbb{Z}$, and a target value $\alpha \in \mathbb{N}$.

Question Is there an assignment β such that the *absolute value* of the sum of the satisfied terms is at least α?

Example

$$\varphi \equiv \underset{\overset{\downarrow}{2}}{(\neg x_1 \wedge x_2 \wedge x_3)} \vee \underset{\overset{\downarrow}{-2}}{(\neg x_1 \wedge x_4)} \vee \underset{\overset{\downarrow}{1}}{(x_1)} \vee \underset{\overset{\downarrow}{-1}}{(x_2 \wedge x_3)}$$

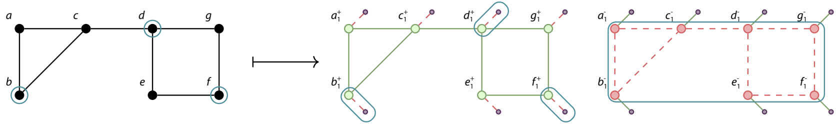
$$\beta = [x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 1]$$

NP-Hardness of ABS-DNF

Theorem

ABS-MONOTONE-DNF *is* NP-*hard.*

Proof Sketch



- This reduction is *not* parameter preserving!
- Is the following problem fixed-parameter tractable?

Problem: $p_{a,d}$-ABS-DNF

Instance	A propositional formula φ in disjunctive normal form with terms of size at most d, a weight function $w\colon \text{terms}(\varphi) \to \mathbb{Z}$, and a target value $a \in \mathbb{N}$.
Parameter	a, d
Question	Is there an assignment β such that the *absolute value* of the sum of the satisfied terms is at least a?

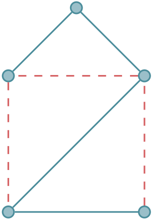
Towards a Parallel Parameterized Algorithm for ABS-DNF

Problem: $\text{p}_{a,d}$-UNBALANCED-HYPERGRAPH

Instance A hypergraph $H = (V, E)$, a weight function $w : E \to \mathbb{Z}$, and $a \in \mathbb{N}$.

Parameter a, d

Question Is there a set $X \subseteq V$ such that $w[X] := \left|\sum_{e\in E, e\subseteq X}\right| \geq a$?

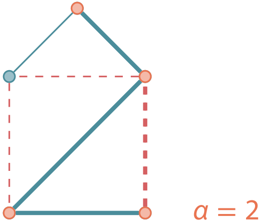


$a = 2$

Towards a Parallel Parameterized Algorithm for ABS-DNF

Problem: $p_{a,d}$-UNBALANCED-HYPERGRAPH

Instance	A hypergraph $H = (V, E)$, a weight function $w: E \to \mathbb{Z}$, and $a \in \mathbb{N}$.
Parameter	a, d
Question	Is there a set $X \subseteq V$ such that $w[X] := \left\| \sum_{e \in E, e \subseteq X} \right\| \geq a$?



A Parallel Kernelization

Rule

Remove all isolated vertices from H.

A Parallel Kernelization

Rule

Remove all isolated vertices from H.

Rule

Remove all edges $e \in E$ with $w(e) = 0$.

A Parallel Kernelization

Rule

Remove all isolated vertices from H.

Rule

Remove all edges $e \in E$ *with* $w(e) = 0$.

Rule (Graph Version)

If $|V| \geq 2a \cdot 2 \cdot \Delta^2$ *return a trivial yes-instance.*

A Parallel Kernelization

Rule

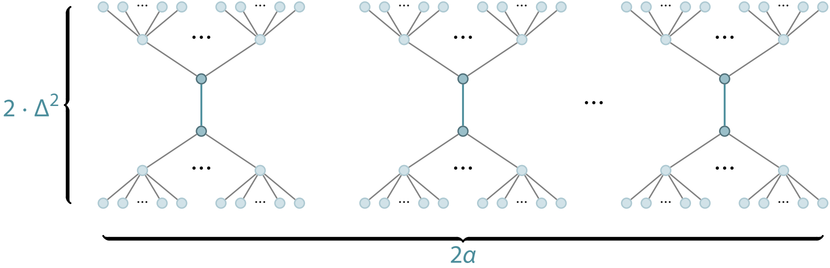
Remove all isolated vertices from H.

Rule

Remove all edges $e \in E$ with $w(e) = 0$.

Rule (Graph Version)

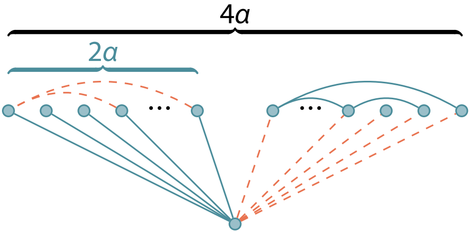
If $|V| \geq 2a \cdot 2 \cdot \Delta^2$ return a trivial yes-instance.



A Parallel Kernelization

Rule (Graph Version)

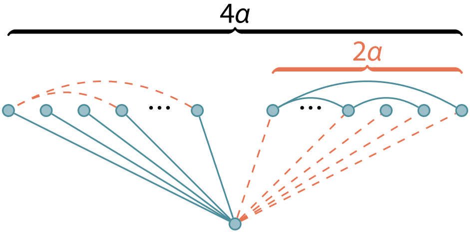
If there is a vertex v with deg$(v) \geq 4a$ return a trivial yes-instance.



A Parallel Kernelization

Rule (Graph Version)

If there is a vertex v with deg$(v) \geq 4a$ return a trivial yes-instance.



A Parallel Kernelization

Rule (Graph Version)

If there is a vertex v with deg$(v) \geq 4a$ *return a trivial yes-instance.*

The rule can be lifted to work with hypergraphs by replacing high-degree vertices with cores of sunflowers.

A Parallel Kernelization

Lemma

All four rules can be implemented in para-AC^0.

A Parallel Kernelization

Lemma

All four rules can be implemented in para-AC^0.

Lemma

Applying the rules yields a kernel for $p_{\alpha,d}$-UNBALANCED-HYPERGRAPH.

A Parallel Kernelization

Lemma

All four rules can be implemented in para-AC^0.

Lemma

Applying the rules yields a kernel for $\mathrm{p}_{a,d}$-UNBALANCED-HYPERGRAPH.

Corollary

$\mathrm{p}_{a,d}$-ABS-DNF $\in$ para-AC^0

Overview of the Results

- The following is known for ABS-DNF.
  - a is the target sum.
  - d the maximum size of a term.

a ╲ d	*constant*	*parameter*	*unbounded*
constant	P	para-AC^0	unknown
parameter	para-AC^0	para-AC^0	W[1]-hard
unbounded	NP-hard	para-NP-hard	NP-hard

Overview of the Results

	para-								
Problem	AC^0	TC^0	$\mathrm{TC}^{0\uparrow}$	L	$\mathrm{L}^{\uparrow}$	$\mathrm{NL}^{\uparrow}$	$\mathrm{TC}^{1\uparrow}$	$\mathrm{AC}^{2\uparrow}$	P
Parameterized by number of clauses to be satisfied (solution size):									
p_k-MAX-SAT	✓								✓
p_g-MAX-SAT-ABOVE-HALF	✗	✓							✓
p_k-MAX-NAE-SAT	✓								✓
$p_{k,d}$-MAX-DNF	✓								✓
Parameterized by number of clauses left unsatisfied (dual parameter):									
p_k-ALMOST-2SAT					✗	✓			✓
p_k-ALMOST-NAE-2SAT					?	✓			✓
p_k-ALMOST-NAE-SAT(2)			✗	✓					✓
p_k-ALMOST-SAT(2)			✗	✓					✓
p_k-ALMOST-DNF	✓								✓
Structural parameter:									
p_{vc}-PARTIAL-MAX-SAT	✗	✓							✓
p_{td}-PARTIAL-MAX-SAT	✗		✓						✓
p_{fvs}-PARTIAL-MAX-SAT			✗				✓		✓
p_{tw}-PARTIAL-MAX-SAT			✗					✓	✓