Bounded Depth Circuits with Weighted Symmetric Gates: Satisfiability, Lower Bounds and Compression*†

Takayuki Sakai[1], Kazuhisa Seto[2], Suguru Tamaki[3], and Junichi Teruyama[4]

1 **Oki Electric Industry Co., Ltd.**
2 **Seikei University, 3-3-1 Kichijoji-Kitamachi, Musashino-shi, Tokyo 180-8633, Japan**
seto@st.seikei.ac.jp
3 **Kyoto University, Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan**
tamak@kuis.kyoto-u.ac.jp
4 **National Institute of Informatics, and JST, ERATO, Kawarabayashi Large Graph Project, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan**
teruyama@nii.ac.jp

Abstract

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *weighted symmetric* if there exist a function $g : \mathbb{Z} \to \{0,1\}$ and integers $w_0, w_1, \ldots, w_n$ such that $f(x_1, \ldots, x_n) = g(w_0 + \sum_{i=1}^{n} w_i x_i)$ holds.

In this paper, we present algorithms for the circuit satisfiability problem of bounded depth circuits with AND, OR, NOT gates and a limited number of weighted symmetric gates. Our algorithms run in time super-polynomially faster than 2^n even when the number of gates is super-polynomial and the maximum weight of symmetric gates is nearly exponential. With an additional trick, we give an algorithm for the maximum satisfiability problem that runs in time $\mathrm{poly}(n^t) \cdot 2^{n-n^{1/O(t)}}$ for instances with n variables, $O(n^t)$ clauses and *arbitrary* weights. To the best of our knowledge, this is the first moderately exponential time algorithm even for Max 2SAT instances with arbitrary weights.

Through the analysis of our algorithms, we obtain average-case lower bounds and compression algorithms for such circuits and worst-case lower bounds for majority votes of such circuits, where all the lower bounds are against the generalized Andreev function. Our average-case lower bounds might be of independent interest in the sense that previous ones for similar circuits with arbitrary symmetric gates rely on communication complexity lower bounds while ours are based on the restriction method.

1998 ACM Subject Classification F.2.0 [Analysis of Algortihms and Problem Complexity] General

Keywords and phrases exponential time algorithm, circuit complexity, circuit minimization, maximum satisfiability

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.82

* Subsumes the technical report [50]. Due to the page limit, we omit many proofs, which can be found in the full version of the paper.

† Supported in part by MEXT KAKENHI (24106003); JSPS KAKENHI (26330011, 26730007, 16H02782); JST, ERATO, Kawarabayashi Large Graph Project; the John Mung Advanced Program of Kyoto University. Part of the work performed while ST was at Department of Computer Science and Engineering, University of California, San Diego and the Simons Institute for the Theory of Computing, Berkeley.

1 Introduction

We are concerned with bounded depth circuits with AND, OR, NOT and (weighted) symmetric gates. Let $\mathbb{Z}$ be the set of integers and $x_1, x_2. \ldots, x_n$ be Boolean variables. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *weighted symmetric* if there exist a function $g : \mathbb{Z} \to \{0,1\}$ and integers $w_0, w_1, \ldots, w_n$ such that $f(x_1, \ldots, x_n) = g(w_0 + \sum_{i=1}^{n} w_i x_i)$ holds. If $w_1 = w_2 = \cdots = w_n = 1$ holds, then f is *symmetric*.

For example, if we set $g(z) = \mathrm{sgn}(z)$, where $\mathrm{sgn}(z) = 1$ if and only if $z \geq 0$, we obtain *majority* functions as symmetric functions and *linear threshold* functions as weighted symmetric functions. If we define $g(z) = 1$ if and only if $z \equiv 0 \bmod m$ for an integer $m \geq 2$, then we obtain *modulo* m functions as symmetric functions.

A (weighted) symmetric gate is a logic gate that computes a (weighted) symmetric function. We denote by $\mathbf{SYM}_w$ the set of weighted symmetric gates such that $\max_i |w_i| \leq w$ holds. When we consider satisfiability and compression algorithms, we assume that $g(z)$ can be evaluated in time polynomial in $\log_2 |z|$, where $|z|$ denotes the absolute value of z. When we consider circuit lower bounds, we assume that g is computable, i.e., there exists a Turing machine that computes g.

1.1 Our contribution

Satisfiability Algorithms: In the *circuit satisfiability problem* (Circuit SAT), our task is, given a Boolean circuit C, to decide whether there exists a 0/1 assignment to the input variables such that C evaluates 1. If input instances are restricted to a class of Boolean circuits $\mathcal{C}$, the problem is called $\mathcal{C}$-SAT. A naïve algorithm can solve Circuit SAT in time $O(\mathrm{poly}(|C|) \cdot 2^n)$, where we denote by $|C|$ the size of C and by n the number of input variables of C respectively. We say an algorithm for $\mathcal{C}$-SAT is *moderately exponential time* if it checks the satisfiability of every $C \in \mathcal{C}$ in time $\mathrm{poly}(|C|) \cdot 2^{n-\omega(\log n)}$, i.e., super-polynomially faster than 2^n. We are interested in for which class $\mathcal{C}$ moderately exponential time satisfiability algorithms exist.

Let $\mathbf{SYM}_w \circ \mathbf{AND}(n, m)$ be the set of n-variate depth 2 circuits with a weighted symmetric gate in $\mathbf{SYM}_w$ at the top and at most m AND gates at the bottom. Let $\mathbf{SYM}_w \circ \mathbf{AC}^0_d(n, m)$ be the set of n-variate unbounded fan-in depth $d+1$ layered circuits with AND, OR, NOT gates and a weighted symmetric gate in $\mathbf{SYM}_w$ such that the top gate is the weighted symmetric gate and each layer contains at most m gates. Let $\mathbf{AC}^0_d[\mathbf{SYM}_w](n, m, t)$ be the set of n-variate unbounded fan-in depth d layered circuits with AND, OR, NOT gates and at most t weighted symmetric gates in $\mathbf{SYM}_w$ such that each layer contains at most m gates.

In this paper, we show moderately exponential time algorithms for the counting version of $\mathcal{C}$-SAT, where $\mathcal{C} \in \{\mathbf{SYM}_w \circ \mathbf{AND}(n, m), \mathbf{SYM}_w \circ \mathbf{AC}^0_d(n, m), \mathbf{AC}^0_d[\mathbf{SYM}_w](n, m, t)\}$, as follows.

▶ **Theorem 1** (depth 2, weighted symmetric gate at the top, AND gates at the bottom)**.** *We can count the number of satisfying assignments for $C \in \mathbf{SYM}_w \circ \mathbf{AND}(n, m)$ deterministically in time*

$$\mathrm{poly}(n, m, \log w) \cdot 2^{n - \Omega\left((n/\log(mw))^{\log n/4\log(nm)}\right)}$$

and exponential space.

The running time is super-polynomially faster than 2^n when, e.g., $m = n^{o(\log n/\log\log n)}$ and $w = 2^{n^{0.99}}$. Note that $\mathbf{SYM}_{2^n}$ contains all Boolean functions (if we ignore the assumption that $g(z)$ can be evaluated in time polynomial in $\log_2 |z|$). The heart of our algorithms is

a (seemingly new) *bottom fan-in reduction* technique inspired by recent developments on the analysis of "greedy restriction" by "concentrated shrinkage" [51, 54, 17, 49]. With an additional trick, we give an algorithm for the maximum satisfiability problem that runs in time $\mathrm{poly}(n^t) \cdot 2^{n-n^{1/O(t)}}$ for instances with n variables, $O(n^t)$ clauses and *arbitrary* weights. To the best of our knowledge, this is the first moderately exponential time algorithm even for Max 2SAT instances with arbitrary weights.

We extend the above algorithm with the help of the depth reduction algorithm due to Beame, Impagliazzo and Srinivasan [7].

▶ **Theorem 2** (depth d, weighted symmetric gate only at the top)**.** *We can count the number of satisfying assignments for $C \in \mathbf{SYM}_w \circ \mathbf{AC}^0_d(n,m)$ deterministically in time*

$$\mathrm{poly}(n,m,\log w) \cdot 2^{n-\Omega\left((n/2^{2d(\log m)^{4/5}}\log(mw))^{\log n/9\log m}\right)}$$

and exponential space.

The running time is super-polynomially faster than 2^n when, e.g., $m = 2^{(\log n/4d)^{5/4}}$ and $w = 2^{n^{0.49}}$.

We further extend the above algorithm relying on the circuit transformation techniques due to Beigel, Reingold and Spielman [9] and Beigel [8].

▶ **Theorem 3** (depth d, $t(n)$ weighted symmetric gates)**.** *We can count the number of satisfying assignments for $C \in \mathbf{AC}^0_d[\mathbf{SYM}_w](n,m,t)$ deterministically in time*

$$\mathrm{poly}(n,m,d,t,\log w) \cdot 2^{n+O(t\log mw)-\Omega\left((n/2^{4d(\log m)^{4/5}}t\log(mw))^{\frac{\log n}{18\log m}}\right)}$$

and exponential space.

The running time is super-polynomially faster than 2^n when, e.g., $m = n^c$, $w = 2^{n^{\frac{1}{40c}}}$ and $t = n^{\frac{1}{40c}}$, where $c \le \frac{\log^{1/4} n}{2(4d)^{5/4}}$.

Although our algorithms run in time super-polynomially faster than 2^n instead of exponentially faster than 2^n ($2^{(1-\varepsilon)n}$ for a universal constant $\varepsilon > 0$), this seems unavoidable due to the Strong Exponential Time Hypothesis (SETH) [12, 32, 34]: The hypothesis states that for all k, there exists $\varepsilon_k > 0$ such that the satisfiability problem of k-CNF formulas cannot be solved in time $2^{(1-\varepsilon_k)n}$. SETH has been used in proving conditional time lower bounds for several exponential time and polynomial time algorithms, see, e.g., [21, 37, 40].

Circuit Lower Bounds: Through the analysis of our satisfiability algorithms, we obtain the following average-case lower bounds.

▶ **Theorem 4** (depth 2, weighted symmetric gate at the top, AND gates at the bottom)**.** *There exists a constant $\alpha > 0$ such that for every m, w and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w}$ such that for every $C \in \mathbf{SYM}_w \circ \mathbf{AND}(n,m)$, it holds that*

$$\Pr_{x\in\{0,1\}^n}[f(x) = C(x)] \le \frac{1}{2} + 2^{-\Omega\left((n/\log(mw))^{\alpha\log n/\log(nm)}\right)}.$$

We also obtain similar average-case lower bounds for $\mathbf{SYM}_w \circ \mathbf{AC}^0_d(n,m)$ and $\mathbf{AC}^0_d[\mathbf{SYM}_w](n,m,t)$, see Theorems 12 and 13 in Section 5.

Our average-case lower bounds might be interesting in the sense that (1) previous ones for similar circuits with arbitrary symmetric gates rely on communication complexity lower

bounds while ours are based on the restriction method and (2) we are not aware of (even worst-case) lower bounds for $\mathbf{SYM}_w \circ \mathbf{AND}$ with $w = n^{\omega(\log n)}$.

Let $\mathcal{C}$ be a set of Boolean circuits and $\mathbf{MAJ} \circ \mathcal{C}$ be the set of Boolean circuits, where $C \in \mathbf{MAJ} \circ \mathcal{C}$ is a majority vote of $\mathcal{C}$ circuits, i.e., $C(x) = \mathrm{sgn}(C_1(x) + \cdots + C_s(x) + w_0)$ holds for some $C_1, \ldots, C_s \in \mathcal{C}$ and an integer w_0.

Combining the above average-case lower bounds and the discriminator lemma due to Hajnal, Maass, Pudlák, Szegedy and Turán [27], we obtain the following worst-case lower bounds.

▶ **Theorem 5** (majority vote of depth 2, weighted symmetric gate at the top, AND gates at the bottom)**.** *There exists a constant $\alpha > 0$ such that for every m, w and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w}$ such that any $C \in \mathbf{MAJ} \circ \mathbf{SYM}_w \circ \mathbf{AND}(n, m)$ cannot compute $f_{n,m,w}$ if the majority gate at the top of C has fan-in at most $2^{o\left((n/\log(mw))^{\alpha \log n/\log(nm)}\right)}$.*

We also obtain similar worst-case lower bounds for $\mathbf{MAJ} \circ \mathbf{SYM}_w \circ \mathbf{AC}^0_d(n, m)$, $\mathbf{MAJ} \circ \mathbf{AC}^0_d[\mathbf{SYM}_w](n, m, t)$ (and $\mathbf{AC}^0_d[\mathbf{SYM}_w](n, m, t)$ with different parameters), see Theorems 24, 25 and 26 in Section 6.

Compression Algorithms: In the *circuit compression problem* (Circuit CMP), our task is, given the truth table of an s-sized Boolean circuit C and an integer $s' \geq s$, to construct a Boolean circuit C' that is at most s'-sized and computes the same function as C. If input instances are restricted to a class of Boolean circuits $\mathcal{C}$, the problem is called $\mathcal{C}$-CMP. In $\mathcal{C}$-CMP, we do not have to construct C' as a circuit in $\mathcal{C}$. Since every n-variate Boolean function can be represented as a $\frac{(1+o(1))2^n}{n}$-sized circuit [39][1], the problem is interesting if $s' \ll 2^n/n$ and in particular we consider the case $s' = 2^{n-\omega(\log n)}$.

A compression algorithm is *efficient* if it runs in time $2^{O(n)}$ given the truth table of an n-variate Boolean function. Note that input length is 2^n and an efficient algorithm runs in polynomial time. The running time analyses of our satisfiability algorithms imply efficient compression algorithms. Let $\mathcal{C} \in \{\mathbf{SYM}_w \circ \mathbf{AND}(n, m), \mathbf{SYM}_w \circ \mathbf{AC}^0_d(n, m), \mathbf{AC}^0_d[\mathbf{SYM}_w](n, m, t)\}$. We obtain deterministic efficient algorithms for $\mathcal{C}$-CMP if parameters n, m, w, d, t are such that the corresponding algorithms for $\mathcal{C}$-SAT run in time $2^{n-\omega(\log n)}$.

1.2 Background

Bounded Depth Circuits with (Weighted) Symmetric Gates: Let $\mathbf{AC}^0$ be the set of bounded depth circuits with AND, OR and NOT gates, $\mathbf{AC}^0[m]$ be the set of $\mathbf{AC}^0$ circuits with modulo m gates, $\mathbf{AC}^0[\mathbf{MAJ}]$ be the set of $\mathbf{AC}^0$ circuits with majority gates (also known as $\mathbf{TC}^0$), $\mathbf{AC}^0[\mathbf{THR}]$ be the set of $\mathbf{AC}^0$ circuits with linear threshold gates and $\mathbf{AC}^0[\mathbf{SYM}_w]$ be the set of $\mathbf{AC}^0$ circuits with gates in $\mathbf{SYM}_w$. Note that for every linear threshold gate, there exists a polynomial size depth 2 majority circuit that computes it [24].

In their seminal work, Razborov [46] and Smolensky [55] showed exponential lower bounds on the size of $\mathbf{AC}^0[m]$ circuits computing majority or mod q functions when m, q are prime powers and relatively prime. Since then, people have been trying to obtain super-polynomial size lower bounds against stronger circuit classes such as $\mathbf{AC}^0[m]$ with arbitrary m or $\mathbf{AC}^0[\mathbf{MAJ}]$. Despite much effort of researchers, super-polynomial size lower bounds have been only shown for such circuit classes with some restriction, see,

[1] Such a representation can be obtained in time $2^{O(n)}$.

e.g., [4, 9, 14, 22, 23, 26, 27, 28] (here we consider circuits computing "explicit" Boolean functions, i.e., functions in NP).

One of the best studied restriction is limiting the number of (weighted) symmetric gates. The following lower bounds are known:

- (Worst-case lower bounds) Exponential lower bounds for $\mathbf{AC}^0[\mathbf{MAJ}]$ circuits with $n^{o(1)}$ majority gates [6, 8] and $\mathbf{AC}^0[\mathbf{THR}]$ circuits with $o(\log n)$ linear threshold gates [44].
- (Average-case lower bounds) super-polynomial lower bounds for $\mathbf{AC}^0[\mathbf{SYM}_1]$ circuits with $o(\log^2 n)$ symmetric gates [58]; arbitrary large polynomial lower bounds for $\mathbf{AC}^0[\mathbf{SYM}_1]$ circuits with $n^{1-o(1)}$ symmetric gates and $\mathbf{AC}^0[\mathbf{THR}]$ circuits with $n^{1/2-o(1)}$ linear threshold gates [38].

The above average-case lower bounds are based on the results of Håstad and Goldmann [29] and Razborov and Wigderson [48] that show average-case lower bounds for $\mathbf{SYM}_1 \circ \mathbf{AND}$ circuits from the communication complexity lower bounds due to Babai, Nisan and Szegedy [5] and also show worst-case lower bounds for $\mathbf{MAJ} \circ \mathbf{SYM}_1 \circ \mathbf{AND}$ circuits using the discriminator lemma.

Circuit Satisfiability: Studying moderately exponential time algorithms for Circuit SAT is motivated by not only the importance in practice, e.g., logic circuit design and constraint satisfaction but also the viewpoint of Boolean circuit complexity. As pointed out by several papers such as [60, 65], there are strong connections between proving circuit lower bounds for $\mathcal{C}$ and designing moderately exponential time algorithms for $\mathcal{C}$-SAT; see also excellent surveys [52, 43, 62]. Typical such connections are:

(1) Some proof techniques such as deterministic/random restriction (shrinkage analysis/switching lemma) simultaneously prove circuit lower bounds for $\mathcal{C}$ and provides $\mathcal{C}$-SAT algorithms [51, 31, 7, 54, 17, 16, 15, 20, 25].

(2) Williams [60, 64] showed that if we obtain a moderately exponential time algorithm for $\mathcal{C}$-SAT and $\mathcal{C}$ satisfies some closure property, then we also have a separation of complexity classes such as $\mathrm{E}^{\mathrm{NP}} \nsubseteq \mathcal{C}$ or $\mathrm{NE} \nsubseteq \mathcal{C}$, where E^{NP} is the set of languages decidable by exponential time Turing machines with NP oracles and NE is the set of languages decidable by non-deterministic exponential time Turing machines; see also [59, 61, 63, 10, 35] for the improvement of such connections. Since then, people have developed moderately exponential time satisfiability algorithms for various circuit classes [33, 18, 30, 1, 3, 2, 42, 19, 57]. In particular, one of the current best lower bounds, $\mathrm{NE} \nsubseteq \mathbf{ACC}^0 \circ \mathbf{THR}$ (also $\mathrm{NE} \nsubseteq \mathbf{ACC}^0 \circ \mathbf{SYM}_1$), was obtained through satisfiability algorithms [63], where $\mathbf{ACC}^0 := \bigcup_m \mathbf{AC}^0[m]$.

Circuit Compression: Circuit CMP is a relaxed version of the circuit minimization problem. Chen, Kabanets, Kolokolova, Shaltiel and Zuckerman [17] established a connection between compression algorithms and circuit lower bounds as follows: If there exists a deterministic efficient algorithm for $\mathcal{C}$-CMP, then $\mathrm{NEXP} \nsubseteq \mathcal{C}$. They also gave efficient compression algorithms for $\mathbf{AC}^0$ circuits, Boolean formulas and branching programs of certain size range. Srinivasan [56] showed an efficient compression algorithm for $\mathbf{AC}^0[m]$ with a prime power m. Carmosino, Impagliazzo, Kabanets and Kolokolova [13] established interesting connections between the tasks of compression/learning and "natural properties" in the sense of Razborov and Rudich [47].

2 Preliminaries

We use random access machines as our computation model. For a set S, we denote by $|S|$ the cardinality of S.

A *literal* is either a Boolean variable or its negation. A *term* is a conjunction of literals. A *Boolean circuit* is a directed acyclic graph whose source nodes are labeled by literals or constants and internal and sink nodes are labeled by logic gates such as AND, OR, NOT, or weighted symmetric gates. A Boolean circuit with a single sink node computes a Boolean function in a natural way. We call source nodes and a sink node *input nodes* and *output node* respectively. The *depth* of a node is defined as the length of the longest path from it to the output node. The *depth* of a Boolean circuit is the maximum value of the depth over all nodes. A Boolean circuit is *layered* if for every edge (u, v), u and v have depth d and $d+1$ for some d.

A Boolean circuit $C : \{0,1\}^n \to \{0,1\}$ is *satisfiable* if there exists a *satisfying assignment* for C, i.e., an assignment $a \in \{0,1\}^n$ such that $C(a) = 1$ holds. For two Boolean functions (or circuits) f, g in the same variables, we write $f \equiv g$ if $f(a) = g(a)$ holds for all $a \in \{0,1\}^n$. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *k-junta* if it depends on at most k variables, i.e., there exist $g : \{0,1\}^k \to \{0,1\}$ and $1 \le i_1 < \cdots < i_k \le n$ such that $f(x_1, \ldots, x_n) = g(x_{i_1}, \ldots, x_{i_k})$ holds.

Let $V = \{x_1, \ldots, x_n\}$. A *restriction* is a mapping $\rho : V \to \{0, 1, *\}$. The meaning of ρ is that if $\rho(x_i) \in \{0,1\}$, then we assign the value $\rho(x_i)$ to x_i, and if $\rho(x_i) = *$, then we leave x_i as it is. Thus, when we *apply* a restriction ρ to a Boolean function f, we obtain the Boolean function $f|_\rho$ defined over the variables $\rho^{-1}(*)$. We also apply a restriction ρ to a Boolean circuit C and obtain a Boolean circuit $C|_\rho$. When we apply a restriction ρ to a Boolean circuit C, we *simplify* a Boolean circuit C using the identities $0 \wedge f \equiv 0$, $1 \wedge f \equiv f$ repeatedly (each appearance of L.H.S. is replaced by R.H.S.).

A *restriction decision tree* T over $x_1, \ldots, x_n$ is an ordinary decision tree except that leaves are not necessarily labeled by 0 or 1. The *height* of T is defined as the number of nodes on the longest path from the root to a leaf and the *size* of T is defined as the number of nodes in T. We identify a path from the root to a leaf with a restriction. A *random root-to-leaf path* is sampled by repeatedly selecting a child of the current node uniformly at random from the root. Note that a path of length ℓ is chosen with probability $2^{-\ell}$.

3 A Dynamic Programming Algorithm for $\mathbf{SYM}_w \circ \mathbf{AND}_k$

We denote by $g \circ \mathbf{AND}_k(n, m, w)$ the set of n-variate Boolean circuits of the form $g(w_0 + \sum_{i=1}^{s} w_i t_i)$, where $g : \mathbb{Z} \to \{0,1\}$, $s \le m$, $w_0, w_1, \ldots, w_s \in \mathbb{Z}, \max_{0 \le i \le s} |w_i| \le w$, and $t_1, \ldots, t_s$ are terms that contain at most k-literals such that $t_i \neq t_j$ holds for $i \neq j$. We define

$$\mathbf{SYM}_w \circ \mathbf{AND}_k(n, m) := \bigcup_{g:\mathbb{Z}\to\{0,1\}} g \circ \mathbf{AND}_k(n, m, w).$$

We specify an element C in $\mathbf{SYM}_w \circ \mathbf{AND}_k(n, m)$ as $C = \{g, w_0, (t_1, w_1), \ldots, (t_s, w_s)\}$ and call s and $\max_{0 \le i \le s} |w_i|$ the *size* and the *maximum weight* of C respectively.

For a restriction ρ, we simplify $C|_\rho = \{g, w_0, (t_1|_\rho, w_1), \ldots, (t_s|_\rho, w_s)\}$ repeatedly if there exists a pair (i, j), $1 \le i < j \le s$ such that $t_i|_\rho \equiv t_j|_\rho$ holds. That is, we delete $(t_j|_\rho, w_j)$ and replace $(t_i|_\rho, w_i)$ by $(t_i|_\rho, w_i + w_j)$. If there are multiple such pairs, we may handle them in arbitrary order.

Our first satisfiability algorithm for $\mathbf{SYM}_w \circ \mathbf{AND}_k(n, m)$ is described in Fig. 1. The algorithm involves two parameters n', m' that are specified in the proof of Theorem 6.

The basic idea is as follows:

Step 1: We construct a table T that contains pairs of the form $(C, \#\mathrm{sat}(C))$ for every circuit C in $g \circ \mathbf{AND}_k(n', m', w')$, where $\#\mathrm{sat}(C)$ denotes the number of satisfying assignments

Algorithm1($C = \{g, w_0, (t_1, w_1), \ldots, (t_s, w_s)\}$: instance, n, m, k, w: integer)
01: **if** $C \notin \mathbf{SYM}_w \circ \mathbf{AND}_k(n, m)$, **return** $\perp$.
02: $T \leftarrow \emptyset$. /* table for dynamic programming */
03: **for each** $C \in g \circ \mathbf{AND}_k(n', m', (s+1) \cdot w)$, /* lexicographical order */
04: $\quad T \leftarrow T \cup \{(C, \#\mathrm{sat}(C))\}$. /* brute force search */
05: $N \leftarrow 0$.
06: **for each** $\rho : V \to \{0, 1, *\}$ such that $\rho^{-1}(*) = \{x_1, \ldots, x_{n'}\}$,
07: $\quad N \leftarrow N + \#\mathrm{sat}(C|_\rho)$. /* binary search in T */
08: **return** N.

Figure 1 A Dynamic Programming Algorithm for $\mathbf{SYM}_w \circ \mathbf{AND}_k$.

for C and n', m', w' are appropriately chosen parameters. Furthermore, pairs are sorted in the lexicographical order with respect to the first coordinate C so that we can use binary search. To do so, we check the number of satisfying assignments for every circuit in $g \circ \mathbf{AND}_k(n', m', w')$ one by one in the lexicographical order using brute force search.

Step 2: Let C be an input instance in $g \circ \mathbf{AND}_k(n, m, w)$. For each restriction ρ that assigns $*$ to the first n' variables of C, we check the number of satisfying assignments for $C|_\rho$ using binary search in T and output the sum of them.

We will show the following theorem.

▶ **Theorem 6.** *We can count the number of satisfying assignments for* $C \in \mathbf{SYM}_w \circ \mathbf{AND}_k(n, m)$ *deterministically in time*

$$\mathrm{poly}(n, m, \log w) \cdot 2^{n - \Omega((n/\log(mw))^{1/k})})$$

and exponential space.

Proof. We denote by $|g \circ \mathbf{AND}_k(n, m, w)|$ the cardinality of $g \circ \mathbf{AND}_k(n, m, w)$. To evaluate the running time of (Step 1), we upper bound the size of the table T using the following fact.

▶ **Fact 7.** *For all* m, *we have*

$$|g \circ \mathbf{AND}_k(n, m, w)| \leq (2w+1)^{\sum_{i=0}^{k} 2^i \binom{n}{i}} \leq 2^{(k+1)(2n)^k \log(2w+1)}.$$

Proof. Note that $\sum_{i=0}^{k} 2^i \binom{n}{i}$ is the number of different terms that consist of at most k-literals (including a constant function 1). Each term has a weight in $\{-w, -w+1, \ldots, w-1, w\}$. Thus, we have the first inequality. The second inequality follows from an elementary calculation. ◀

Thus, we can bound the running time of Lines 03-04 from above by

$$2^{(k+1)(2n')^k \log(2(m+1)w+1)} \times \mathrm{poly}(m', \log(mw)) \cdot 2^{n'},$$

where we set $m' = \sum_{i=0}^{k} 2^i \binom{n'}{i} \leq (k+1)(2n')^k$.

Next we evaluate the running time of (Step 2). Note that the following guarantees that every $C|_\rho$ in Line 06 belongs to $g \circ \mathbf{AND}_k(n', m', (m+1) \cdot w)$.

▶ **Fact 8.** *Let* $C = \{g, w_0, (t_1, w_1), \ldots, (t_m, w_m)\}$. *If* $C \in g \circ \mathbf{AND}_k(n, m, w)$ *holds, then for all restriction* ρ *with* $|\rho^{-1}(*)| = n'$, *we have* $C|_\rho \in g \circ \mathbf{AND}_k(n', m', (m+1) \cdot w)$.

Proof. By the definition of $\mathbf{SYM}_w \circ \mathbf{AND}_k(n,m)$, we have $\sum_{i=0}^{s} |w_i| \leq (m+1)w$. This implies the maximum weight of $C|_\rho$ is at most $(m+1)w$. ◀

For each $C|_\rho$, binary search in Line 07 takes time at most

$$\log_2 |g \circ \mathbf{AND}_k(n', m', (m+1)\cdot w)| \times \mathrm{poly}(m', \log(mw)) = \mathrm{poly}(m', \log(mw)).$$

Thus, we can bound the running time of Lines 06-07 above by

$$\mathrm{poly}(m, m', \log(mw)) \cdot 2^{n-n'}.$$

If we set $n' = \left(\frac{n}{(k+1)2^{k+1}\log(2(m+1)w+1)}\right)^{1/k} = \Theta((n/\log(mw))^{1/k})$, the total running time of **Algorithm1** is bounded from above by $\mathrm{poly}(n, m, \log w) \cdot 2^{n-\Omega((n/\log(mw))^{1/k})}$. This completes the proof. ◀

▶ Remark. In the case when $g(z) = \mathrm{sgn}(z)$, we can reduce the weight of the top gate of $C|_\rho$ from $(m+1)w$ to $2^{n'^{O(k)}}$ efficiently by Theorem 16 in [41]. With this trick, we can handle Max SAT instances with arbitrary weights.

4 A Greedy Restriction Algorithm for $\mathbf{SYM}_w \circ \mathbf{AND}_k$

For a term t, we denote by $|t|$ the width of t, i.e., the number of literals in t and by $\mathrm{var}(t)$ the set of variables that appear in t (possibly negated). Let $C \in \mathbf{SYM}_w \circ \mathbf{AND}_k(n,m)$ be a circuit $\{g, w_0, (t_1, w_1), \ldots, (t_s, w_s)\}$. We define $\mathrm{var}_\ell(C) := \cup_{i:|t_i|\geq\ell}\mathrm{var}(t_i)$, $\mathrm{freq}_\ell(C,x) := |\{t_i \in C \mid x \in \mathrm{var}(t_i), |t_i| \geq \ell\}|$, and $L_\ell(C) := \sum_{i:|t_i|\geq\ell} |t_i|$.

Our second satisfiability algorithm for $\mathbf{SYM}_w \circ \mathbf{AND}_k(n,m)$ is described in Fig. 2. The basic idea is as follows:

Step 1: Choose a positive integer ℓ according to the input. We seek for a variable, say x, that occurs most frequently in terms of width at least ℓ. We recursively run the algorithm for $C|_{x=0}$ and $C|_{x=1}$. Here $C|_{x=a}$ denotes the circuit obtained from C by applying a restriction ρ such that $\rho(x) = a \in \{0,1\}$ and $\rho(x') = *$ for $x' \neq x$.

Step 2: If there is no term of width at least ℓ, we call **Algorithm1**.

We will show the following theorem which implies Theorem 1 by setting $k = n$.

▶ **Theorem 9.** *We can count the number of satisfying assignments for* $C \in \mathbf{SYM}_w \circ \mathbf{AND}_k(n,m)$ *deterministically in time*

$$\mathrm{poly}(n, m, \log w) \cdot 2^{n-\Omega\left((n/\log(mw))^{\log n/4\log(km)}\right)}$$

and exponential space.

Proof. Let us define a sequence of random variables $\{C_i\}$ inductively as $C_0 := C$ and $C_{i+1} := C_i|_{x=a}$, where $x = \arg\max_{x\in\mathrm{var}(C_i)} \mathrm{freq}_\ell(C_i, x)$ and a is a uniform random bit.

We can think of the computation of **Algorithm2** as a rooted binary tree. That is, the root node is labeled with C_0, the left and right children of the root are labeled with $C_0|_{x=0}$ and $C_0|_{x=1}$, and so on. Then, if we pick a node of depth $n - n'$ uniformly at random, the distribution of its label is identical to that of the random variable $C_{n-n'}$.

We would like to bound the running time of **Algorithm2**$(C_{n-n'}, n', n', \ell)$. It is obviously bounded from above by $\mathrm{poly}(n, m, \log w) \cdot 2^{n'}$. Furthermore, if $L_\ell(C_{n-n'}) < \frac{n'}{2}$ holds, the

Algorithm2($C = \{g, w_0, (t_1, w_1), \ldots, (t_s, w_s)\}$: **instance,** n, n', ℓ: **integer**)
01: **if** $n > n'$,
02: $x = \arg\max_{x \in \mathrm{var}(C)} \mathrm{freq}_\ell(C, x)$.
03: $N_0 \leftarrow$ **Algorithm2**($C|_{x=0}, n-1, n', \ell$).
04: $N_1 \leftarrow$ **Algorithm2**($C|_{x=1}, n-1, n', \ell$).
05: **return** $N_0 + N_1$.
06: **else**
07: $N \leftarrow 0$.
08: **for each** $\rho : \mathrm{var}(C) \to \{0, 1, *\}$ such that $\rho^{-1}(\{0,1\}) = \mathrm{var}_\ell(C)$,
09: $w' \leftarrow$ the maximum weight of $C|_\rho$.
10: $N \leftarrow N +$ **Algorithm1**($C|_\rho, n - |\mathrm{var}_\ell(C)|, m', \ell - 1, w'$).
11: **return** N.

Figure 2 A Greedy Restriction Algorithm for $\mathbf{SYM}_w \circ \mathbf{AND}_k$.

running time can be bounded by $2^{n'/2}\times$ (the running time of **Algorithm1**$(C', n'/2, m', \ell - 1, w')$) for $C' \in \mathbf{SYM}_{w'} \circ \mathbf{AND}_{\ell-1}(n'/2, m')$ with $m' = \ell \cdot (n')^{\ell-1}$ and $w' = (m+1)w$. We need the following lemma.

▶ **Lemma 10** (Greedy bottom fan-in reduction). *Let $C \in \mathbf{SYM}_w \circ \mathbf{AND}_k(n, m)$. For all $n' \geq 4$, we have*

$$\Pr\left[L_\ell(C_{n-n'}) \geq 2^\ell \cdot L_\ell(C) \cdot \left(\frac{n'}{n}\right)^{\frac{\ell+2}{2}}\right] < 2^{-n'}.$$

Since $L_\ell(C) \leq km$, if we set $n' = \frac{1}{16}\left(\frac{n}{km}\right)^{2/\ell} \cdot n$ in the above lemma, we have

$$2^\ell \cdot L_\ell(C) \cdot \left(\frac{n'}{n}\right)^{\frac{\ell+2}{2}} \leq \frac{n'}{2},$$

that is, we have $L_\ell(C_{n-n'}) < n'/2$ with probability at least $1 - 2^{-n'}$. If we set $\ell = \frac{4\log(km)}{\log n}$, then the total running time of **Algorithm2** is bounded from above by the sum of

$$\mathrm{poly}(n, m, \log w) \cdot 2^{n-n'} \cdot 2^{-n'} \cdot 2^{n'}$$

and

$$\mathrm{poly}(n, m, \log w) \cdot 2^{n-n'} \cdot (1 - 2^{-n'}) \cdot 2^{n'/2} \cdot 2^{n'/2 - \Omega((n'/(\log(m'w'))^{1/\ell})}$$

according to whether $L_\ell(C_{n-n'}) \geq n'/2$ holds or not. An elementary calculation completes the proof. ◀

▶ Remark. The novelty of our algorithm and its analysis is a new way of reducing the bottom fan-in of circuits in a greedy manner. Intuitively, given a $\mathbf{SYM}_w \circ \mathbf{AND}_k$ circuit with m gates, greedy restriction produces a collection of $\mathbf{SYM}_{w'} \circ \mathbf{AND}_{k'}$ circuits with $k' = O(\log(km)/\log n)$ such that at least one of the circuits in the collection is satisfiable if and only if so is the original circuit. Note that previous techniques such as Schuler's width reduction [53, 11] or the standard random restriction achieve $k' = O(\log(m/n))$ and this bound is not sufficient for our purpose.

5 Average-Case Circuit Lower Bounds

Through the analysis of our satisfiability algorithms, we obtain the following average-case lower bounds.

▶ **Theorem 11** (depth 2, weighted symmetric gate at the top, AND gates at the bottom)**.** *There exists a constant $\alpha > 0$ such that for every m, w and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w}$ such that for every $C \in \mathbf{SYM}_w \circ \mathbf{AND}(n, m)$, it holds that*

$$\Pr_{x\in\{0,1\}^n}[f_{n,m,w}(x) = C(x)] \leq \frac{1}{2} + 2^{-\Omega\left((n/\log(mw))^{\alpha \log n/\log(nm)}\right)}.$$

▶ **Theorem 12** (depth d, weighted symmetric gate only at the top)**.** *There exists a constant $\alpha > 0$ such that for every m, w, d and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w,d}$ such that for every $C \in \mathbf{SYM}_w \circ \mathbf{AC}^0_d(n, m)$, it holds that*

$$\Pr_{x\in\{0,1\}^n}[f_{n,m,w,d}(x) = C(x)] \leq \frac{1}{2} + 2^{-\Omega\left((n/2^{2d(\log m)^{4/5}}\log(mw))^{\alpha \log n/\log m}\right)}.$$

▶ **Theorem 13** (depth d, $t(n)$ weighted symmetric gates)**.** *There exists a constant $\alpha > 0$ such that for every m, w and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w,d,t}$ such that for every $C \in \mathbf{AC}^0_d[\mathbf{SYM}_w](n, m, t)$, it holds that*

$$\Pr_{x\in\{0,1\}^n}[f_{n,m,w,d,t}(x) = C(x)] \leq \frac{1}{2} + 2^{-\Omega\left((n/2^{2d(\log m')^{4/5}}\log(m'w'))^{\alpha \log n/\log m'}\right)},$$

where $m' = m2^{t+1}$ and $w' = (mw)^{2^{t+1}}$.

In the rest of this section, we give a proof of Theorem 11. The proof of Theorem 12 is similar and we omit proof. Theorem 13 immediately follows from Theorem 12 with the idea of the proof of Theorem 5.1 in [8].

5.1 Generalized Andreev function

In this section, we review the construction of average-case hard Boolean functions due to [17, 36]. We begin with some definitions.

▶ **Definition 14** (Statistical distance)**.** Two distributions X, Y over a set E are *ε-close* if $|\Pr[X \in A] - \Pr[Y \in A]| \leq \varepsilon$ holds for every $A \subseteq E$.

▶ **Definition 15.** A set $A \subseteq \{0,1\}^n$ is a *subcube of dimension k* if there exist $1 \leq i_1 < \cdots < i_k \leq n$ and $a_{i_1}, \ldots, a_{i_k} \in \{0,1\}$ such that $A = \{x \in \{0,1\}^n \mid x_{i_1} = a_{i_1}, \ldots, x_{i_k} = a_{i_k}\}$.

▶ **Definition 16** (Bit-fixing extractor)**.** A function $f : \{0,1\}^n \to \{0,1\}^m$ is an *(n, k, m, ε)-bit-fixing extractor* if $f(X)$ and the uniform distribution over $\{0,1\}^m$ are ε-close for every distribution X that is uniform over a subcube of $\{0,1\}^n$ of dimension at least k.

We need the following explicit construction due to Rao.

▶ **Lemma 17** (Efficient bit-fixing extractor [45])**.** *There exist constants $\alpha, \beta > 0$ such that for every $k \geq (\log n)^\alpha$, there exists a polynomial time computable $\mathrm{Ext}_{n,k} : \{0,1\}^n \to \{0,1\}^m$ that is an (n, k, m, ε)-bit-fixing extractor with $m = 0.9k$ and $\varepsilon \leq 2^{-k^\beta}$.*

We also need an efficient and explicit construction of list decodable codes.

▶ **Definition 18** (List-Decodable Code)**.** A function $f : \{0,1\}^k \to \{0,1\}^n$ is (p, L)-list-decodable if $|\{y \in \{0,1\}^k \mid \Delta(f(x), f(y)) \leq pn\}| \leq L$ holds for every $x \in \{0,1\}^k$, where $\Delta(a, b)$ denotes the Hamming distance between a and b.

▶ **Lemma 19** (Efficient List-Decodable Code (Folklore), see Theorem 6.4 in [17])**.** *There exists a function* $\mathrm{Enc}_{n,r} : \{0,1\}^{4n} \to \{0,1\}^{2^r}$ *that is* (p, L)*-list-decodable with* $p = 1/2 - O(2^{-r/4})$ *and* $L = O(2^{r/2})$*. Furthermore, there exists an algorithm that, given* $x \in \{0,1\}^{4n}$ *and* $z \in \{0,1\}^{2^r}$*, computes* $(\mathrm{Enc}_{n,r}(x))_z$ *in polynomial time.*

We are ready to define the average-case hard Boolean functions: The generalized Andreev function $A_{n,k} : \{0,1\}^{4n} \times \{0,1\}^n \to \{0,1\}$ is defined as $A_{n,k}(x, y) := (\mathrm{Enc}_{n,0.9k}(x))_{\mathrm{Ext}_{n,k}(y)}$. Let $K(x)$ denote the *Kolmogorov complexity* of a string $x \in \{0,1\}^*$. The following lemma plays an important role in the proofs of our average-case lower bounds.

▶ **Lemma 20** (Theorem 6.5 in [17])**.** *There exist constants* $\alpha, \gamma > 0$ *such that the following holds. Let* $k \geq (\log n)^\alpha$ *and* C *be a* k*-variate circuit whose binary description length is at most* n *in a some fixed encoding scheme. Let* $\rho : \{x_1, \ldots, x_n\} \to \{0, 1, *\}$ *be a restriction with* $|\rho^{-1}(*)| = k$*. Fix* $a \in \{0,1\}^{4n}$ *with* $K(a) \geq 3n$ *and define* $f(y) := A_{n,k}(a, y)$*. Then, we have*

$$\Pr_{y' \in \{0,1\}^k}[C(y') = f|_\rho(y')] \leq \frac{1}{2} + \frac{1}{2^{k^\gamma}}.$$

The following fact can be shown by a counting argument.

▶ **Fact 21.** *For every* $0 < p < 1$, $\mathbf{Pr}_{x \in \{0,1\}^n}[K(x) \leq (1-p)n] \leq 2^{-pn+1}$.

5.2 Proof of Theorem 11

Fix n, m, w and let $n' = (n/\log(mw))^{\log n/4 \log(nm)}$. Select any $a \in \{0,1\}^{4n}$ with $K(a) \geq 3n$ and let $f(y) := A_{n,n'}(a, y)$. We show the following lemma.

▶ **Lemma 22.** *For every* $C \in \mathbf{SYM}_w \circ \mathbf{AND}(n, m)$*, it holds that*

$$\Pr_{y \in \{0,1\}^n}[C(y) = f(y)] \leq \frac{1}{2} + 2^{-\Omega(n'^\gamma)},$$

where $\gamma > 0$ *is a universal constant from Lemma 20.*

Assuming this, the proof of Theorem 11 is complete since by Fact 21, we have

$$\begin{aligned}
\Pr_{x,y}[A_{n,n'}(x, y) = C(x, y)] &\leq \Pr_x[K(x) < 3n] + \Pr_x[K(x) \geq 3n] \\
&\times \Pr_{x,y}[A_{n,n'}(x, y) = C(x, y) \mid K(x) \geq 3n] \\
&\leq 2^{-\Omega(n)} + \max_{x : K(x) \geq 3n} \Pr_y[A_{n,n'}(x, y) = C(x, y)] \\
&\leq 2^{-\Omega(n)} + \frac{1}{2} + 2^{-\Omega(n'^\gamma)}.
\end{aligned}$$

.

Proof of Lemma 22. We can see that from the proofs of Theorems 6 and 9, C can be computed by a restriction decision tree T of height $n - n'$ such that (1) each leaf is labeled by a circuit in $\mathbf{SYM}_{w'} \circ \mathbf{AND}_{k'}(n', m')$ for some m', k', w' and (2) except for a $2^{-n^{\Omega(1)}}$ fraction of leaves, such a circuit can be described by using at most n bits (due to Fact 7). Let $\sigma(C)$

denote the description length of a circuit C in a fixed encoding scheme. Let ρ be a random restriction sampled by selecting a leaf of T uniformly at random and y_ρ be a uniform random element of $\{0,1\}^{\rho^{-1}(*)}$. Then, we have

$$\begin{aligned}\Pr_y[C(y) = f(y)] &\leq \Pr_\rho[\sigma(C|_\rho) > n] + \Pr_\rho[\sigma(C|_\rho) \leq n] \\ &\times \Pr_{\rho,y_\rho}[C|_\rho(y_\rho) = f|_\rho(y_\rho) \mid \sigma(C|_\rho) \leq n] \leq 2^{-n^{\Omega(1)}} + \frac{1}{2} + 2^{-\Omega(n'^\gamma)},\end{aligned}$$

where the last inequality is by Item (2) above and Lemma 20. This completes the proof. ◀

6 Worst-Case Lower Bounds

From the average-case lower bounds in Section 5, we obtain the following worst-case lower bounds.

▶ **Theorem 23** (majority vote of depth 2, weighted symmetric gate at the top, AND gates at the bottom)**.** *There exists a constant $\alpha > 0$ such that for every m, w and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w}$ such that $C \in \mathbf{MAJ} \circ \mathbf{SYM}_w \circ \mathbf{AND}(n,m)$ cannot compute $f_{n,m,w}$ if the majority gate at the top of C has fan-in at most $2^{o\left((n/\log(mw))^{\alpha \log n/\log(nm)}\right)}$.*

▶ **Theorem 24** (majority vote of depth d, weighted symmetric gate only at the top)**.** *There exists a constant $\alpha > 0$ such that for every m, w, d and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w,d}$ such that any $C \in \mathbf{MAJ} \circ \mathbf{SYM}_w \circ \mathbf{AC}^0_d(n,m)$ cannot compute $f_{n,m,w,d}$ if the majority gate at the top of C has fan-in at most $2^{o\left((n/2^{2d(\log m)^{4/5}} \log(mw))^{\alpha \log n/\log m}\right)}$.*

▶ **Theorem 25** (majority vote of depth d, $t(n)$ weighted symmetric gates)**.** *There exists a constant $\alpha > 0$ such that for every m, w, d, t and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w,d,t}$ such that any $C \in \mathbf{MAJ} \circ \mathbf{AC}^0_d[\mathbf{SYM}_w](n,m,t)$ cannot compute $f_{n,m,w,d,t}$ if the majority gate at the top of C has fan-in at most $2^{o\left((n/2^{2d(\log m')^{4/5}} \log(m'w'))^{\alpha \log n/\log m'}\right)}$, where $m' = m2^{t+1}$ and $w' = (mw)^{2^{t+1}}$.*

▶ **Theorem 26** (depth d, $t(n)$ weighted symmetric gates)**.** *There exists a constant $\alpha > 0$ such that for every m, w, d, t and sufficiently large n, there exists a polynomial time computable function $f_{n,m,w,d,t}$ such that any $C \in \mathbf{AC}^0_d[\mathbf{SYM}_w](n,m,t)$ cannot compute $f_{n,m,w,d,t}$ if*

$$t = o\left((n/2^{2d(\log m')^{4/5}} \log(m'w'))^{\alpha \log n/\log m'}\right)$$

holds, where $m' = m(t+1)$ and $w' = m^t w^{t+1}$.

We need a corollary of the discriminator lemma.

▶ **Lemma 27** (Discriminator Lemma [27])**.** *If a circuit $C \in \mathbf{MAJ} \circ \mathcal{C}$ is a majority vote of k circuits $C_1, \ldots, C_k \in \mathcal{C}$, then for some $1 \leq i \leq k$, we have*

$$|\Pr_x[C_i(x) = 1 \mid C(x) = 1] - \Pr_x[C_i(x) = 1 \mid C(x) = 0]| \geq \frac{1}{k}.$$

For $f, g : \{0,1\}^n \to \{0,1\}$, let $\mathrm{Corr}(f,g) := |\mathbf{Pr}_x[f(x) = g(x)] - \mathbf{Pr}_x[f(x) \neq g(x)]|$.

▶ **Corollary 28.** *For $\varepsilon \geq 0$, if C in Lemma 27 also satisfies that*

$$|\Pr_x[C(x) = 0] - \Pr_x[C(x) = 1]| = 2\varepsilon,$$

then we have $\mathrm{Corr}(f, g) \geq \frac{1}{k} - 2\varepsilon$.

Theorems 23, 24 and 25 immediately follow from Theorems 11, 12 and 13 with Corollary 28. Theorem 26 can be shown by combining the relation of circuit classes, Theorem 12 and Corollary 28.

References

1. Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 218–230, 2015.
2. Kazuyuki Amano and Atsushi Saito. A nonuniform circuit class with multilayer of threshold gates having super quasi polynomial size lower bounds against NEXP. In *Proceedings of the 9th International Conference on Language and Automata Theory and Applications (LATA)*, pages 461–472, 2015.
3. Kazuyuki Amano and Atsushi Saito. A satisfiability algorithm for some class of dense depth two threshold circuits. *IEICE Transactions*, 98-D(1):108–118, 2015.
4. James Aspnes, Richard Beigel, Merrick L. Furst, and Steven Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):135–148, 1994.
5. László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.
6. David A. Mix Barrington and Howard Straubing. Complex polynomials and circuit lower bounds for modular counting. *Computational Complexity*, 4:325–338, 1994.
7. Paul Beame, Russell Impagliazzo, and Srikanth Srinivasan. Approximating AC^0 by small height decision trees and a deterministic algorithm for $\#\mathrm{AC}^0$ SAT. In *Proceedings of the 27th Conference on Computational Complexity (CCC)*, pages 117–125, 2012.
8. Richard Beigel. When do extra majority gates help? polylog(n) majority gates are equivalent to one. *Computational Complexity*, 4:314–324, 1994.
9. Richard Beigel, Nick Reingold, and Daniel A. Spielman. PP is closed under intersection. *J. Comput. Syst. Sci.*, 50(2):191–202, 1995.
10. Eli Ben-Sasson and Emanuele Viola. Short PCPs with projection queries. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP), Part I*, pages 163–173, 2014.
11. Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity (CCC)*, pages 252–260, 2006.
12. Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Revised Selected Papers from the 4th International Workshop on Parameterized and Exact Computation (IWPEC)*, pages 75–85, 2009.
13. Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Algorithms from natural lower bounds. In *Proceedings of the 31st Conference on Computational Complexit (CCC)*, pages 10:1–10:24, 2016.
14. Arkadev Chattopadhyay and Kristoffer Arnsfelt Hansen. Lower bounds for circuits with few modular and symmetric gates. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 994–1005, 2005.
15. Ruiwen Chen. Satisfiability algorithms and lower bounds for Boolean formulas over finite bases. In *Proceedings of the 40th International Symposium on Mathematical Foundations of Computer Science (MFCS), Part II*, pages 223–234, 2015.

16 Ruiwen Chen and Valentine Kabanets. Correlation bounds and #SAT algorithms for small linear-size circuits. In *Proceedings of the 21st International Conference on Computing and Combinatorics (COCOON)*, pages 211–222, 2015.

17 Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining circuit lower bound proofs for meta-algorithms. *Computational Complexity*, 24(2):333–392, 2015.

18 Ruiwen Chen, Valentine Kabanets, and Nitin Saurabh. An improved deterministic #SAT algorithm for small De Morgan formulas. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS), Part II*, pages 165–176, 2014.

19 Ruiwen Chen and Rahul Santhanam. Improved algorithms for sparse MAX-SAT and MAX-k-CSP. In *Proceedings of the 18th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 33–45, 2015.

20 Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. In *Proceedings of the 31st Conference on Computational Complexit (CCC)*, pages 1:1–1:35, 2016.

21 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. In *Proceedings of the 27th Annual IEEE Conference on Computational Complexity (CCC)*, pages 74–84, 2012.

22 Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels Schmitt, and Hans-Ulrich Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 171–182, 2001.

23 Mikael Goldmann. On the power of a threshold gate at the top. *Inf. Process. Lett.*, 63(6):287–293, 1997.

24 Mikael Goldmann and Marek Karpinski. Simulating threshold circuits by majority circuits. *SIAM J. Comput.*, 27(1):230–246, 1998.

25 Alexander Golovnev, Alexander S. Kulikov, Alexander Smal, and Suguru Tamaki. Circuit size lower bounds and #SAT upper bounds through a general framework. In *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2016, to appear.

26 Parikshit Gopalan and Rocco A. Servedio. Learning and lower bounds for AC^0 with threshold gates. In *Proceedings of the 13th APPROX and the 14th RANDOM*, pages 588–601, 2010.

27 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993.

28 Kristoffer Arnsfelt Hansen and Peter Bro Miltersen. Some meet-in-the-middle circuit lower bounds. In *Proceedings of the 29th International Symposium Mathematical Foundations of Computer Science (MFCS)*, pages 334–345, 2004.

29 Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

30 Russell Impagliazzo, Shachar Lovett, Ramamohan Paturi, and Stefan Schneider. 0-1 integer linear programming with a linear number of constraints. *Electronic Colloquium on Computational Complexity (ECCC)*, TR14-24, 2014.

31 Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for AC^0. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 961–972, 2012.

32 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

33 Russell Impagliazzo, Ramamohan Paturi, and Stefan Schneider. A satisfiability algorithm for sparse depth two threshold circuits. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 479–488, 2013.

34 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

35 Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP), Part I*, pages 749–760, 2015.

36 Ilan Komargodski, Ran Raz, and Avishay Tal. Improved average-case lower bounds for demorgan formula size. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 588–597, 2013.

37 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

38 Shachar Lovett and Srikanth Srinivasan. Correlation bounds for poly-size AC^0 circuits with $n^{1-o(1)}$ symmetric gates. In *Proceedings of the 14th APPROX 2011 and the 15th RANDOM*, pages 640–651, 2011.

39 Oleg Borisovich Lupanov. On a method of circuit synthesis (in Russian). *Izvestiâ vysših učebnyh zavedenij, Radiofiz*, 1:120–140, 1958.

40 Dániel Marx. Consequences of SETH: Tight bounds for some more problems, 2015. (abstract, slides and archived video). URL: `https://simons.berkeley.edu/talks/daniel-marx-2015-09-04`.

41 Saburo Muroga, Iwao Toda, and Satoru Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271(5):376–418, 1961.

42 Atsuki Nagao, Kazuhisa Seto, and Junichi Teruyama. A moderately exponential time algorithm for k-IBDD satisfiability. In *Proceedings of the 14th International Symposium, on Algorithms and Data Structures (WADS)*, pages 554–565, 2015.

43 Igor Carboni Oliveira. Algorithms versus circuit lower bounds. *Electronic Colloquium on Computational Complexity (ECCC)*, TR13-117, 2013.

44 Vladimir V. Podolskii. Exponential lower bound for bounded depth circuits with few threshold gates. *Inf. Process. Lett.*, 112(7):267–271, 2012.

45 Anup Rao. Extractors for low-weight affine sources. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC)*, pages 95–101, 2009.

46 Alexander Razborov. Lower bounds on the size of bounded-depth networks over a complete basis with logical addition. *Mathematical Notes of the Academy of Sci. of the USSR*, 41(4):333–338, 1987.

47 Alexander Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

48 Alexander Razborov and Avi Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Inf. Process. Lett.*, 45(6):303–307, 1993.

49 Takayuki Sakai, Kazuhisa Seto, and Suguru Tamaki. Solving sparse instances of Max SAT via width reduction and greedy restriction. *Theory Comput. Syst.*, 57(2):426–443, 2015.

50 Takayuki Sakai, Kazuhisa Seto, Suguru Tamaki, and Junichi Teruyama. A satisfiability algorithm for depth-2 circuits with a symmetric gate at the top and AND gates at the bottom. *Electronic Colloquium on Computational Complexity (ECCC)*, TR15-136, 2015.

51 Rahul Santhanam. Fighting perebor: New and improved algorithms for formula and QBF satisfiability. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 183–192, 2010.

52 Rahul Santhanam. Ironic complicity: Satisfiability algorithms and circuit lower bounds. *Bulletin of the EATCS*, 106:31–52, 2012.

53 Rainer Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *J. Algorithms*, 54(1):40–44, 2005.

54 Kazuhisa Seto and Suguru Tamaki. A satisfiability algorithm and average-case hardness for formulas over the full binary basis. *Computational Complexity*, 22(2):245–274, 2013.

55 Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 77–82, 1987.

56 Srikanth Srinivasan. A compression algorithm for $\mathrm{AC}^0[\oplus]$ circuits using certifying polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, TR15-142, 2015.

57 Avishay Tal. #SAT algorithms from shrinkage. *Electronic Colloquium on Computational Complexity (ECCC)*, TR15-114, 2015.

58 Emanuele Viola. Pseudorandom bits for constant-depth circuits with few arbitrary symmetric gates. *SIAM J. Comput.*, 36(5):1387–1403, 2007.

59 Fengming Wang. NEXP does not have non-uniform quasipolynomial-size ACC circuits of $o(\log \log n)$ depth. In *Proceedings of the 8th Annual Conference on Theory and Applications of Models of Computation (TAMC)*, pages 164–170, 2011.

60 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM J. Comput.*, 42(3):1218–1244, 2013.

61 Ryan Williams. Natural proofs versus derandomization. In *Proceedings of the 45th ACM Symposium on Theory of Computing Conference (STOC)*, pages 21–30, 2013.

62 Ryan Williams. Algorithms for circuits and circuits for algorithms. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC)*, pages 248–261, 2014.

63 Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *Proceedings of the 46th Symposium on Theory of Computing (STOC)*, pages 194–202, 2014.

64 Ryan Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2, 2014.

65 Francis Zane. *Circuits, CNFs, and satisfiability.* PhD thesis, UC San Diego, 1998.