Electronic Colloquium on Computational Complexity, Revision 3 of Report No. 192 (2018)

Circuit Depth Reductions[*]

Alexander Golovnev[†] Alexander S. Kulikov[‡] R. Ryan Williams[§]

Abstract

The best known size lower bounds against unrestricted circuits have remained around $3n$ for several decades. Moreover, the only known technique for proving lower bounds in this model, gate elimination, is inherently limited to proving lower bounds of less than $5n$. In this work, we propose a non-gate-elimination approach for obtaining circuit lower bounds, via certain depth-three lower bounds. We prove that every (unbounded-depth) circuit of size s can be expressed as an OR of $2^{s/3.9}$ 16-CNFs. For DeMorgan formulas, the best known size lower bounds have been stuck at around $n^{3-o(1)}$ for decades. Under a plausible hypothesis about probabilistic polynomials, we show that $n^{4-\varepsilon}$-size DeMorgan formulas have $2^{n^{1-\Omega(\varepsilon)}}$-size depth-3 circuits which are approximate sums of $n^{1-\Omega(\varepsilon)}$-degree polynomials over $\mathbb{F}_2$. While these structural results do not immediately lead to new lower bounds, they do suggest new avenues of attack on these longstanding lower bound problems.

Our results complement the classical depth-3 reduction results of Valiant, which show that *logarithmic*-depth circuits of linear size can be computed by an OR of $2^{\varepsilon n}$ n^{δ}-CNFs, and slightly stronger results for series-parallel circuits. It is known that no purely graph-theoretic reduction could yield interesting depth-3 circuits from circuits of super-logarithmic depth. We overcome this limitation (for small-size circuits) by taking into account both the graph-theoretic and functional properties of circuits and formulas.

We show that improvements of the following pseudorandom constructions imply super-linear circuit lower bounds for log-depth circuits via Valiant's reduction: dispersers for varieties, correlation with constant degree polynomials, matrix rigidity, and hardness for depth-3 circuits with constant bottom fan-in. On the other hand, our depth reductions show that even modest improvements of the known constructions give elementary proofs of improved (but still linear) circuit lower bounds.

[*] The work of Ryan Williams is supported by NSF CCF-1909429 and CCF-1741615. The work of Alexander S. Kulikov presented in Section 4 is supported by the RNF grant 18-71-10042.

[†] Georgetown University, email: alexgolovnev@gmail.com

[‡] Steklov Institute of Mathematics at St. Petersburg and St. Petersburg State University, email: alexanderskulikov@gmail.com

[§] MIT CSAIL & EECS, email: rrw@mit.edu

ISSN 1433-8092

1 Introduction

The Boolean circuit model is natural for computing Boolean functions. A circuit corresponds to a simple straight line program where every instruction performs a binary operation on two operands, each of which is either an input or the result of a previous instruction. The structure of this program is extremely simple: no loops, no conditional statements. Still, we know no functions in P (or even NP, or even E^{NP}) that requires even $3.1n$ binary instructions ("size") to compute on inputs of length n. This is in sharp contrast with the fact that it is easy to *non-constructively* find such functions: simple counting arguments show a random function on n variables has circuit size $\Omega(2^n/n)$ with probability $1 - o(1)$ [Sha49].

The strongest known circuit size lower bound $(3 + \frac{1}{86})n - o(n)$ was proved for affine dispersers for sublinear dimension [FGHK16]. This proof, as well as all previous proofs for general circuit lower bounds against explicit functions, is based on the method of gate elimination. The main idea is to find a substitution to an input variable that eliminates sufficiently many gates from the given circuit, and then proceed by induction. While this is the most successful method known so far for proving lower bounds for unrestricted circuits, the resulting case analysis becomes increasingly tedious: when eliminating (say) 3 or 4 gates, one must consider all possible cases when two of these gates coincide. It is difficult to imagine a proof of $5n$ lower bound using these ideas. This intuition was recently made formal in [GHKK18], where it was shown that a certain formalization of the gate elimination technique is unable to obtain a stronger than $5n$ lower bound. Therefore we must find new approaches for proving lower bounds against circuits of unbounded depth. Let us review some of the prior results on various circuit models.

Linear Circuits. Superlinear lower bounds are not known even for linear circuits, i.e., circuits consisting of only XOR gates (also known as $\oplus$ gates). Note every linear function with one output has a circuit of size at most $n - 1$. For linear circuits, we consider *linear transformations*, multi-output functions of the form $f(x) = Ax$ where $A \in \mathbb{F}_2^{m \times n}$. For a random matrix $A \in \{0,1\}^{n \times n}$, the size of the smallest linear circuit computing Ax is $\Theta(n^2/\log n)$ [Lup56] with probability $1 - o(1)$, but for explicitly-constructed matrices the strongest known lower bound is $3n - o(n)$ due to Chashkin [Cha94]. Interestingly, Chashkin's proof is not based on gate elimination: he first shows that the parity check matrix $H \in \{0,1\}^{\log n \times n}$ of the Hamming code has circuit size $2n - o(n)$ by proving that every circuit for H has at least $n - o(n)$ gates of out-degree at least 2.[1] Then he "pads" H to an $n \times n$ matrix H' and shows that $n - o(n)$ additional gates are needed for H'. Similarly, the best known lower bound on the complexity of linear circuits with $\log n \le m < o(n^2)$ outputs is $2n + m - o(n)$ (also follows from [Cha94]).

Log-Depth Circuits. Nothing stronger than a $(3 + \frac{1}{86})n - o(n)$ size lower bound is known even for circuits of depth $O(\log n)$. It is straightforward to show that any function that depends on all of its n variables requires depth at least $\log n$. One can also present an explicit function that cannot be computed by a circuit of depth smaller than $2\log n - o(\log n)$ using Nechiporuk's lower bound of $n^{2-o(1)}$ on formula size over the full binary basis [Nec66]. Still, proving superlinear size lower bounds for circuits of depth $O(\log n)$ remains a major open problem [Val77].

[1] All logarithms are base 2 unless noted otherwise.

Constant-Depth Circuits. Another natural and simple model of computation is bounded-depth unbounded fan-in circuits, which correspond to highly parallelizable computation. In this paper, we focus on depth-2 circuits of the form AND $\circ$ OR (i.e., CNFs) and depth-3 circuits of the form OR $\circ$ AND $\circ$ OR (i.e., ORs of CNFs), where the inputs of the circuit are variables and their negations, and the gates have unbounded fan-in. Such circuits are much more structured, and therefore are easier to analyze and to prove lower bounds. For example, it is easy to show that the minimal number of clauses in a CNF computing the parity of n bits is equal to 2^{n-1}, which yields an optimal lower bound for depth-2 circuits. However, already for depth 3 there is a large gap between known lower and upper bounds: it is known [Dan96, Ser18] that the minimum depth-3 circuit size of a random function on n variables is $\Theta(2^{n/2})$, but the best known lower bound for an explicit function is $2^{\Omega(\sqrt{n})}$ [Hås86, HJP93, PPZ97, Bop97, PPSZ05, MW17].

Much stronger lower bounds are known for depth-3 circuits where the fan-in of the "bottom" gates (those closest to the inputs) is bounded by a parameter k. Namely, for any $k \leq O(\sqrt{n})$, Paturi, Saks, and Zane [PPZ97] proved a $2^{n/k}$ lower bound for computing parity, Wolfovitz [Wol06] proved a lower bound of $(1+1/k)^{n+O(\log n)}$ for $\mathrm{ETHR}_{\frac{n}{k+1}}$ [2], and a stronger lower bound of $2^{\frac{\mu_k n}{k-1}}$ for $k \geq 3$ and some constants $\mu_k > 1$ was proven in [PPSZ05] for a BCH code. For example, [PPSZ05] gives a lower bound of $2^{0.612n}$ when the bottom fan-in of the circuit is $k = 3$, and a lower bound of $2^{n/10}$ for the bottom fan-in $k = 16$. For the case of bottom fan-in $k = 2$, even a $2^{n-o(n)}$ lower bound is known [PSZ97].

A simple counting argument shows that for any constant $k = O(1)$, a random function requires depth-3 circuits of size $2^{n-o(n)}$. Calabro, Impagliazzo, and Paturi [CIP06] construct a family of $2^{O(n^2)}$ explicit functions, most of which require depth-3 circuits with $k = O(1)$ of size $2^{n-o(n)}$. Santhanam and Srinivasan [SS12] improve on this by constructing such a family of functions of size $2^{f(n)}$ for every $f(n) = \omega(n \log n)$.

DeMorgan Formulas. While explicit super-linear lower bounds for *circuits* are not known, there are super-linear lower bounds for *formulas*. In this paper, we focus on the well-studied DeMorgan formulas, which are circuits where every intermediate computation is used exactly once: all gates have out-degree one, and the operations are fan-in two ANDs and ORs, with inputs being variables and their negations. The two most successful methods for proving lower bounds on DeMorgan formula size are random restrictions [Sub61, And87, IN93, PZ93, Hås98, Tal14] as well as Karchmer–Wigderson games and the Karchmer–Raz–Wigderson conjecture [Khr71, KW90, KRW95, GMWW14, DM16]. Both approaches have led to a lower bound of $n^{3-o(1)}$ and are currently stuck at giving stronger lower bounds.

1.1 Valiant's Depth Reduction

Remarkably, a classical result of Valiant from the 70's relates three of the four models above: linear, log-depth, and constant-depth circuits. Using a depth reduction for DAGs [EGS75], Valiant [Val77] shows that for any circuit of size cn and depth d, and for every integer k, one can remove at most $\frac{2ckn}{\log d}$ wires such that the resulting circuit has depth at most $d/2^k$. Letting k be a sufficiently large constant, this wire-removal lemma shows how any circuit of size $O(n)$ and depth $O(\log n)$ can be converted into an OR $\circ$ AND $\circ$ OR circuit where the OR output gate has fan-in $2^{O(n/\log\log n)}$ and the lower OR gates have fan-in $O(n^{\varepsilon})$ for any desired $\varepsilon > 0$. Hence, by exhibiting a function that

[2] $\mathrm{ETHR}_{\frac{n}{k+1}}$ outputs 1 if and only if the sum of the n input bits over the integers equals $\frac{n}{k+1}$.

has no depth-3 circuit with these restrictions, it follows that this function cannot be computed by circuits of linear size and logarithmic depth. Unfortunately, the best known lower bounds on depth-3 circuits (as mentioned earlier) are still too far from those required for this reduction.

In the same paper, Valiant introduced the notion of matrix rigidity (a similar notion was independently introduced by Grigoriev [Gri76]) and related it to the size of linear circuits of log-depth using ideas similar to those described above. Alas, the known lower bounds on matrix rigidity are also far from being able to give new lower bounds on the size of log-depth linear circuits.

1.2 Our Results: New Depth Reductions

The main contributions of this paper are new reductions to depth-3 circuits that work for *unrestricted* circuits and (conditionally) for super-cubic formulas, as well as new results connecting various pseudorandom objects to circuit lower bounds. In particular, we show how to express super-cubic DeMorgan formulas as subexponential-size depth-3 circuits of a certain form, under the hypothesis that DeMorgan formulas have probabilistic polynomials of non-trivial degree. This suggests an approach for improving formula size lower bounds, by proving strong lower bounds on depth-3 circuits.

1.2.1 Depth Reductions for Circuits

In Valiant's depth reduction, one can only have $d/2^k < \log n$ (and $< cn$ removed edges) for circuits of depth $d \leq O(\log n)$. Thus, Valiant's depth reduction technique does not yield interesting results for circuits of super-logarithmic depth. Moreover, Schnitger and Klawe [Sch82, Sch83, Kla94] construct an explicit family of DAGs showing that the parameters achieved by Valiant are essentially optimal. Their counterexamples convincingly show that a pure graph-theoretic approach to circuit depth reduction cannot give non-trivial results for unrestricted circuits.

In this paper, we overcome this difficulty by presenting a counterpart of Valiant's depth reduction that works for circuits of unrestricted depth. Our depth reduction takes into account not only the underlying graph of a circuit, but also the *functions* computed by the circuit gates.

Our first result shows that unbounded-depth circuits of size less than $3.9n$ can be converted into $2^{\delta n}$ disjunctions of short 16-CNFs, for some $\delta < 1$.

Theorem 1.1. *Every circuit of size s can be computed as an $OR_{2^{\lceil \frac{s}{2} \rceil}} \circ AND_s \circ OR_2$ circuit and as an $OR_{2^{\lceil \frac{s}{3.9} \rceil}} \circ AND_{2^{14} \cdot s} \circ OR_{16}$ circuit.*

As a consequence, in order to prove a $3.9n - o(n)$ size lower bound on *unrestricted* circuits, it suffices to provide a function that cannot be computed by an OR of fewer than $2^{n-o(n)}$ 16-CNF's. To prove Theorem 1.1, we gradually transform the given circuit into an OR of CNF's by carefully picking a suitable internal gate and branching on its two possible output values. In contrast to Valiant's reduction, our transformation works for circuits of arbitrary depth. This is achieved by an argument that takes into account both the graph structure of the circuit *and* the functional properties of the gates involved. Since in this approach we can branch on *internal* gates (inside the circuit), we can avoid a massive case analysis. This also distinguishes our approach from known circuit lower bound proofs based on gate elimination, which must set input gates (or gates very close to the inputs) for the argument to work.

It should be noted that known satisfiability algorithms based on branching, as well as circuit lower bounds based on gate elimination [PPZ97, PPSZ05, Sch05, San10, CK15] may be viewed as depth-reductions for small circuits: if at most k variables are set in any branch before the circuit has a "trivial" form, then the circuit can be expressed as an OR of 2^k "trivial" forms. At the same time, the known techniques in this line of work appear stuck at lower bounds of around $3n$, and provably cannot go beyond linear-size bounds [GHKK18].

On the way to proving Theorem 1.1, we study structural results about converting small circuits into disjunctions of k-CNFs, that have curious connections to properties of k-CNFs found in the Satisfiability Coding Lemma [PPZ97, PPSZ05] and Sparsification Lemma [IPZ01, CIP06]. In particular, we ask the following question.

Open Problem 1.1. *Prove or disprove: for any constant c, any circuit of size cn can be computed as an*

$$OR_{2^{(1-\delta(c))n}} \circ AND \circ OR_{\gamma(c)}$$

circuit, for some $\delta(c) > 0$ and integer $\gamma(c) \geq 1$.

If such depth-3 circuits always existed, this would constitute a new approach to proving super-linear circuit lower bounds. If no depth-3 circuit of this form exists for some linear-size circuits, then we would have a separation between linear-size circuits and (for example) super-linear-size series-parallel circuits (by Valiant's reduction for such circuits, see Theorem 2.1). Note that for the gate elimination method such limitations are known [GHKK18], and they do not apply to the approach presented in this work.

Our second result is a new "non-rigidity" result for matrices with small linear circuits: if a matrix M over $\mathbb{F}_2$ can be computed by a linear circuit of size s, then it is possible to flip at most 16 bits in every row of M to drop its rank below $s/4$. This opens up an approach to proving linear circuit lower bounds on sizes up to $4n$.

Theorem 1.2. *For every matrix $M \in \mathbb{F}_2^{m\times n}$ of linear circuit complexity s, $\mathbb{R}_M(\lfloor s/4 \rfloor) \leq 16$.*

1.2.2 Pseudorandom Objects and Circuit Lower Bounds

The classical result by Valiant shows that improvements of known depth-3 circuit lower bounds and rigid matrices imply super-linear log-depth circuit lower bounds. Our depth reductions show that even modest improvements of the known constructions also give modest improvements of unrestricted circuit lower bounds.

In Section 5, we show that Valiant's and our reduction are applicable to two more types of pseudorandom objects: dispersers for varieties, and functions having small correlation with low degree polynomials. These implications are briefly summarized[3] in Table 1.

1.2.3 Depth Reductions for Formulas

For DeMorgan formulas we give a conditional depth-reduction (stated informally, see Theorem 3.4 for a formal statement): if there is an $\varepsilon > 0$ such that DeMorgan formulas of size s have probabilistic polynomials of degree $s^{1-\varepsilon}$ and error $1/3$ over $\mathbb{F}_2$, then for some $\delta > 0$ every DeMorgan formula of size $O(n^{3+\delta})$ can be written as an approximate sum of $2^{n^{1-\gamma}}$ degree-$n^{1-\gamma}$ $\mathbb{F}_2$-polynomials for

[3] In this table we only present strongest implications from the strongest premises. Our reductions would still give new circuit lower bounds even from weaker objects (see Section 5 for formal statements of the results). For example, the second line of the table says that a lower bound of $2^{n-o(n)}$ against depth-3 circuits would give a lower bound of $3.9n$. On the other hand, a lower bound of $2^{0.8n}$ would lead to an elementary proof of a lower bound of $3.1n$.

	improving known lower bound	to lower bound	implies lower bound
V	$s_3^{n^\varepsilon}(f) \geq 2^{n^{1-\varepsilon}}$ [PPZ97]	$s_3^{n^\varepsilon}(f) \geq 2^{\omega\left(\frac{n}{\log\log n}\right)}$	$s_{\log}(f) = \omega(n)$
*	$s_3^{16}(f) \geq 2^{\frac{n}{10}}$ [PPSZ05]	$s_3^{16}(f) \geq 2^{n-o(n)}$	$s(f) \geq 3.9n$
V	$\left(n^\varepsilon, \infty, 2^{n-n^{1/2-\varepsilon}}\right)$-disp. [Rem16]	$\left(n^\varepsilon, \infty, 2^{n-\omega\left(\frac{n}{\log\log n}\right)}\right)$-disp.	$s_{\log}(f) = \omega(n)$
*	$\left(16, \infty, 2^{(1-\varepsilon)n}\right)$-disp. [VW08]	$(16, 1.3n, 2^{o(n)})$-disp.	$s(f) \geq 3.9n$
*	$\left(16, \frac{n}{(\log n)^c}, 2^{o(n)}\right)$-disp. [CT15]	$(16, 1.3n, 2^{o(n)})$-disp.	$s(f) \geq 3.9n$
V	$\mathbb{R}_M\left(\omega\left(\frac{n}{\log\log n}\right)\right) > \log\log n$ [Fri93]	$\mathbb{R}_M\left(\omega\left(\frac{n}{\log\log n}\right)\right) > n^\varepsilon$	$s_{\oplus,\log}(M) = \omega(n)$
*	$\mathbb{R}_M(\frac{n}{65}) > 16$ [PV91]	$\mathbb{R}_M(n - o(n)) > 16$	$s_\oplus(M) \geq 4n$

Table 1: Comparing the depth reductions of this paper (labeled with *) with the depth reduction of Valiant [Val77] (labeled with V). We use the following notation (all formal definitions are given in Sections 2 and 5): $s(f)$ is the smallest size of a circuit computing f, $s_{\log}$ refers to circuits of depth $O(\log n)$, s_3^k refers to circuits that are ORs of k-CNFs, $s_\oplus$ refers to circuits consisting of $\oplus$ gates only; (d, m, s)-disp. stands for a (d, m, s)-disperser, a function that is not constant on any subset of the Boolean hypercube of size at least s that is defined as the set of common roots of at most m polynomials of degree at most d; $\mathbb{R}_M(r)$ is the row-rigidity of M for the rank r over $\mathbb{F}_2$, i.e., the smallest row-sparsity of a matrix A such that $\operatorname{rank}(M \oplus A) \leq r$.

a constant $\gamma > 0$.[4] Moreover, if there are probabilistic polynomials of degree $O(\sqrt{s})$ for DeMorgan formulas of size s (which we conjecture is true), our depth reduction holds for DeMorgan formulas of size $n^{3.99}$.

Interestingly, the techniques used to express DeMorgan formulas as depth-3 circuits are totally different from those used in Theorem 1.1 and 1.2. Namely, we first balance a formula (without increasing its size too much), decompose it into a small top part and several small bottom formulas, approximate the top part by a real-valued low-degree polynomial, then rewrite the bottom parts as probabilistic polynomials (as hypothesized). Finally, we collapse these two polynomials into a depth-3 circuit.

The hypothesis that lower-degree probabilistic polynomials exist for every DeMorgan formula of size s looks very plausible. We have not found an example of a size-s formula that resists the construction of an $O(\sqrt{s})$-degree probabilistic polynomial. Note that such polynomials do exist in the real-approximation sense [Rei11]. For example, every symmetric function (such as MAJORITY) has probabilistic polynomials of $O(\sqrt{s})$ degree [AW15], and it is not hard to show that the layered OR-AND tree of depth $\log_2(s)$ has a probabilistic polynomial of $O(\sqrt{s})$ degree as well; in fact, *any* layered tree of depth $\log_2(s)$ with the same gate type at each layer (AND or OR) has such degree.[5] It is possible that there are "nasty" formulas that resist lower-degree probabilistic polynomials, but given the examples we already know, we do not know what they might look like.

[4] Similar results can be stated for $\mathbb{F}_p$ where p is any prime.

[5] Briefly: we can always write such formulas as either an OR of ANDs of $O(\sqrt{s})$ literals, or an AND of ORs of $O(\sqrt{s})$ literals. From there, we can simply replace the output gate with an $O(1)$-degree probabilistic polynomial (as in Razborov [Raz87]), and the other gates with exact polynomials of $O(\sqrt{s})$ degree.

Open Problem 1.2. *Prove or disprove: every DeMorgan formula of size s has a probabilistic polynomial over $\mathbb{F}_2$ of degree $O(\sqrt{s})$ with constant error less than $1/2$.*

1.3 Motivating Example

Here we provide a simple example of a reduction of unbounded circuits to depth-3 circuits, to give an idea of what is possible.

A *formula* is a circuit where every internal gate (i.e. not the inputs and not the output) has out-degree exactly 1. In our simple example, we will show that a circuit of size, say, $2.7n$ can be computed by an OR of $2^{0.9n}$ formulas of small size ($2.7n$). Since we know almost-quadratic lower bounds [Nec66] on formula size, we may hope to find a function which is not computable by an OR of $\ll 2^n$ linear-size formulas.

Lemma 1.3 (Toy Example)**.** *Every circuit of size s can be expressed as an OR of $2^{\lceil s/3 \rceil}$ formulas, each of size less than s.*

Proof. For a circuit C, let $s(C)$ denote its size. For $s \leq 3$, we just transform a circuit into a single formula of the same size. For $s > 3$, we proceed by induction. If the given circuit $\mathcal{C}$ is a formula, no transformation is needed. Otherwise take the topologically first gate G of out-degree at least 2. Note G is computed by a formula (all previous gates have out-degree 1); let $t = s(G)$ be the size of this formula. Consider two minimum-size circuits $\mathcal{C}_0$ and $\mathcal{C}_1$ that compute the same function as $\mathcal{C}$ on the input sets $\{x \in \{0,1\}^n : G(x) = 0\}$ and $\{x \in \{0,1\}^n : G(x) = 1\}$, respectively. We claim that $s(\mathcal{C}_0), s(\mathcal{C}_1) \leq s - t - 2 \leq s - 3$, since to compute $\mathcal{C}_0$ and $\mathcal{C}_1$ one can remove the subcircuit in C computing gate G as well as two successors of G. The successors can be removed because G outputs a constant on both parts of the considered partition of the Boolean hypercube, and all gates in the subcircuit of G are only needed to compute G (G is computed by a formula). Now, note that

$$\mathcal{C}(x) \equiv (\neg G(x) \wedge \mathcal{C}_0(x)) \vee (G(x) \wedge \mathcal{C}_1(x))\,.$$

Applying the induction hypothesis to $\mathcal{C}_0$ and $\mathcal{C}_1$, we can rewrite $\mathcal{C}$ as an OR of at most $2^{\lceil (s-3)/3+1 \rceil} \leq 2^{\lceil s/3 \rceil}$ formulas of size $(s - t - 2) + (t + 1) < s$. □

This result would imply a circuit lower bound of $3n - o(n)$ for any function that has correlation at most $2^{-n+o(n)}$ with all formulas of linear size. While we do know functions that have exponentially small correlation $2^{-\varepsilon n}$ with formulas of linear size [San10, KLP12, ST13, KRT13, Tal14, IK17], none of them gives a bound of $2^{-n+o(n)}$. At any rate there is an inherent limitation for this toy approach. By Parseval's identity, every Boolean function has a Fourier coefficient $\geq 2^{-n/2}$. This implies that the correlation of this function with the corresponding parity function is at least $2^{-n/2}$ (and this is essentially tight correlation with small formulas for a random function). Since every parity on a subset of inputs can be computed by a formula of size $\leq n$, Lemma 1.3 would only be able to prove circuit lower bounds of $1.5n$.

In order to prove stronger circuit lower bounds, we need to improve both parameters: the constant 3 in the exponent, and the class of formulas we reduce circuits to. Our Theorem 1.1 achieves this: it reduces a circuit to an OR of $2^{\lceil \frac{s}{3.9} \rceil}$ formulas, each of which is a 16-CNF. Therefore strong enough correlation bounds against 16-CNFs would yield new circuit lower bounds.

2 Definitions and Preliminaries

2.1 Unrestricted Circuits

Let $B_{n,m}$ be the set of all Boolean functions $f\colon \{0,1\}^n \to \{0,1\}^m$ and let $B_2 = B_{2,1}$. A *circuit* is a directed acyclic graph that has n nodes of in-degree 0 labeled with $x_1, \ldots, x_n$ that are called *input gates*. All other nodes are called *internal gates*, have in-degree 2, and are labeled with operations from B_2. Some m gates are also marked as output gates. Such a circuit computes a function from $B_{n,m}$ in a natural way. The *size* $s(\mathcal{C})$ of a circuit $\mathcal{C}$ is its number of *internal* gates. This definition extends naturally to functions: $s(f)$ is the smallest size of a circuit computing the function f.

The *depth* of a gate G is the maximum number of edges (also called *wires*) on a path from an input gate to G. The depth of a circuit is the maximum depth of its gates. By $s_{\log n}(f)$ we denote the smallest size of a circuit of depth $O(\log n)$ computing f.

A circuit is called *linear* if it consists of $\oplus$ gates only. The corresponding circuit size measure is denoted by $s_{\oplus}$.

Our unrestricted circuits are usually drawn with input gates at the top, so by a top gate of a circuit we mean a gate that is fed by two variables.

2.2 Series-Parallel Circuits

A *labeling* of a directed acyclic graph $G = (V, E)$ is a function $\ell\colon V \to \mathbb{N}$ such that for every edge $(u, v) \in E$ one has $\ell(u) < \ell(v)$. A graph/circuit G is called *series-parallel* if there exists a labeling ℓ such that for no two edges $(u, v), (u', v') \in E$, $\ell(u) < \ell(u') < \ell(v) < \ell(v')$. The corresponding circuit complexity measure is s_{sp}.

2.3 Depth-3 Circuits

Unlike unrestricted circuits, depth-3 circuits are usually drawn the other way around, i.e., with the output gate at the top. In this paper, we focus on $\text{OR} \circ \text{AND} \circ \text{OR}$ circuits, i.e., ORs of CNFs. We will use subscripts to indicate the fact that the fan-in of a particular layer is bounded. Namely, an $\text{OR}_p \circ \text{AND}_q \circ \text{OR}_r$ circuit is an OR of at most p CNFs each of which contains at most q clauses and at most r literals in every clause. Since the gates of a depth 3 circuit are allowed to have an unbounded fan-in, it is natural to define the size of such a circuit as its number of wires. It is not difficult to see that for $k = O(1)$ the size of an $\text{OR} \circ \text{AND} \circ \text{OR}_k$ circuit is equal to the fan-in of its output gate up to a polynomial factor in n. By $s_3^k(f)$ we denote the smallest size of an $\text{OR} \circ \text{AND} \circ \text{OR}_k$ circuit computing f.

2.4 Rigidity

We say that a matrix $M \in \mathbb{F}_2^{m \times n}$ is *s-sparse* if each *row* of M contains at most s non-zero elements. The *rigidity* of a matrix $M \in \mathbb{F}_2^{m \times n}$ for the rank parameter r is the minimum sparsity of a matrix $A \in \{0,1\}^{m \times n}$ such that $\operatorname{rank}_{\mathbb{F}_2}(M \oplus A) \le r$:

$$\mathbb{R}_M(r) = \min\{s\colon \operatorname{rank}_{\mathbb{F}_2}(M \oplus A) \le r,\ A \text{ is } s\text{-sparse}\}\,.$$

2.5 Probabilistic, Approximate, and Robust Polynomials

Since even functions of small circuit and formula complexity may only have large-degree polynomial representations, it often proves convenient to use randomized polynomials or polynomials which approximate (rather than exactly compute) a given function.

Definition 2.1 (Probabilistic polynomials)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. A distribution $\mathcal{D}$ of n-variate degree-d polynomials over $\mathbb{F}_2$ is a* probabilistic polynomial *for f with degree d and error ε if for every $x \in \{0,1\}^n$,*

$$\Pr_{p \sim \mathcal{D}}[f(x) = p(x)] \geq 1 - \varepsilon.$$

Definition 2.2 (Approximate Polynomials)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. An n-variate multilinear degree-d polynomial p over $\mathbb{R}$ is an* approximate polynomial *for f with degree d and error ε if for every $x \in \{0,1\}^n$,*

$$|p(x) - f(x)| \leq \varepsilon.$$

Definition 2.3 (Robust Polynomials)**.** *Let $f\colon \{0,1\}^n \to [0,1]$ be a polynomial over $\mathbb{R}$. Then a polynomial $p\colon \mathbb{R}^n \to \mathbb{R}$ is* δ-robust *for f if for every $x \in \{0,1\}^n$ and for every $\varepsilon \in [-1/3, 1/3]^n$,*

$$|f(x) - p(x + \varepsilon)| \leq \delta.$$

2.6 Valiant's Depth Reductions

Here we formally recall the classical depth reduction results by Valiant [Val77].

Theorem 2.1 ([Val77, Cal08, Vio09])**.** *For every $c \geq 1$ and $\varepsilon > 0$ there exists a $\delta > 0$ such that every circuit $\mathcal{C}$ of size cn and depth $c \log n$ can be computed as*

1. *an $OR_{2^{\frac{\delta n}{\log\log n}}} \circ AND \circ OR_{n^{\varepsilon}}$ circuit*
2. *and as an $OR_{2^{\varepsilon n}} \circ AND \circ OR_{2^{(\log n)^{1-\delta}}}$ circuit.*

Furthermore, for every $c \geq 1$ and $\varepsilon > 0$ there is a $k \geq 1$ such that every series-parallel circuit of size cn and unbounded depth can be computed as an $OR_{2^{\varepsilon n}} \circ AND \circ OR_k$ circuit.

Theorem 2.1 applied to linear circuits yields the following.

Theorem 2.2 ([Val77, Cal08, Vio09])**.** *Let $M \in \mathbb{F}^{m \times n}$ be a matrix. For every $c \geq 1$ and $\varepsilon > 0$ there exists $\delta > 0$ such that, if a linear circuit $\mathcal{C}$ of size cn and depth $c \log n$ computes Mx for every $x \in \mathbb{F}^n$, then*

1. *$\mathbb{R}_M\left(\frac{\delta n}{\log\log n}\right) \leq n^{\varepsilon}$;*
2. *and $\mathbb{R}_M(\varepsilon n) \leq 2^{(\log n)^{1-\delta}}$.*

Furthermore, for every $c \geq 1$ and $\varepsilon > 0$ there is a $k \geq 1$ such that if $\mathcal{C}$ is a series-parallel linear circuit of size cn and unbounded depth, then $\mathbb{R}_M(\varepsilon n) \leq k$.

3 Formula Depth Reduction

In this section, we give a (conditional) depth reduction for DeMorgan formulas. We start by balancing a given formula. For this we use the following result due to Tal [Tal14].

Lemma 3.1 (Claim VI.2 in [Tal14])**.** *Let F be a DeMorgan formula of size s over the set of variables $X = \{x_1, ..., x_n\}$, and t be some parameter; then, there exist $k \leq 36s/t$ formulas over X, denoted by $T_1, ..., T_k$, each of size at most t, and there exists a read-once formula F' of size k such that $F'(T_1(x), ..., T_k(x)) = F(x)$ for all $x \in \{0,1\}^n$.*

Below we will also make use of the following results by Reichardt [Rei11] and Sherstov [She12].

Theorem 3.2 ([Rei11])**.** *If $f\colon \{0,1\}^n \to \{0,1\}$ can be computed by a DeMorgan formula of size s, then f has an approximate polynomial of degree $O(\sqrt{s})$ with error $\varepsilon = 1/10$.*

Theorem 3.3 ([She12])**.** *If $f\colon \{0,1\}^n \to [0,1]$ is a polynomial of degree d over $\mathbb{R}$, then there is a δ-robust polynomial p for f of degree $O(d + \log(1/\delta))$.*

Now we are ready to present the main result of this section: Assuming DeMorgan formulas of size s have probabilistic polynomials of degree $O(s^{1-\delta})$ for some $\delta > 0$, we will obtain subexponential-size depth-3 circuits computing formulas of super-cubic size.

In the following, a SUM gate will compute an *approximate sum*: a (real-weighted) sum of the inputs such that, over all Boolean inputs, the sum is within $\pm 1/3$ of the 0-1 value of a desired Boolean function.

Theorem 3.4. *Suppose for some $\delta > 0$, DeMorgan formulas of size ℓ have probabilistic polynomials of degree $\ell^{1-\delta}$ with error $1/3$. Then for every $\alpha < \delta/(1-\delta)$ there is a $\gamma > 0$, so that for every formula F of size $s = O(n^{3+\alpha})$, there is a $2^{n^{1-\gamma}}$-size approximate sum of degree-$n^{1-\gamma}$ $\mathbb{F}_2$-polynomials computing F. That is, F can be computed by a*

$$\mathsf{SUM}_{2^{n^{1-\gamma}}} \circ \mathsf{MOD2}_{2^{n^{1-\gamma}}} \circ \mathsf{AND}_{n^{1-\gamma}} .$$

Proof. First, we apply Lemma 3.1 to F for some parameter t to be defined later. We obtain a read-once formula F' of size $k = O(s/t)$, and k formulas $T_1, \ldots, T_k$ each of size $\leq t$.

Let p be an approximate polynomial (over the reals) for F' of degree $d = O(\sqrt{k})$ with error $1/10$, guaranteed by Theorem 3.2. Applying Theorem 3.3, we get a $1/10$-robust polynomial p' for p of degree $d' = O(\sqrt{k})$.

By the hypothesis of the theorem, we know that each T_i has a probabilistic polynomial of degree $O(t^{1-\delta})$ with error $\varepsilon = 1/3$. For each T_i, draw $O(\log s)$ independent copies of this probabilistic polynomial, and take their majority vote with an $O(\log s)$-degree polynomial. For an appropriate leading constant in the big-O, we can obtain a probabilistic polynomial for T_i of degree $O(t^{1-\delta} \cdot \log s)$ with error $1/(10s)$.

Let $\mathcal{D}_1, \ldots, \mathcal{D}_k$ be probabilistic polynomials of degree $D = O(t^{1-\delta} \cdot \log s)$ with error $\varepsilon = 1/(10s)$ for the formulas $T_1, \ldots, T_k$. The error bound $\varepsilon = 1/(10s)$ guarantees that for every $x \in \{0,1\}^n$, *all* k polynomials compute the correct value with probability at least $9/10$.

Now for every T_i, we compute the average A_i (over the reals) of $O(n)$ independent samples from $\mathcal{D}_i$. By a Chernoff bound and union bound, each A_i is within $\pm 1/10$ of the correct 0-1 value for T_i, over all 2^n inputs x, with probability of error $1/\exp(n)$. By the properties of robust polynomials,

p' fed the sums A_i will still output the correct value (within $\pm 1/10$) for *all* inputs $x \in \{0,1\}^n$, for some choice of samples.

Therefore F can be computed by a

$$\mathsf{SUM}_{n^{d'}} \circ \mathsf{PRODUCT}_{d'} \circ \mathsf{SUM}_{O(n)} \circ \mathsf{MOD2} \circ \mathsf{AND}_D.$$

Applying distributivity to the PRODUCT of SUMs, we get

$$\mathsf{SUM}_{n^{d'}} \circ \mathsf{SUM}_{n^{O(d')}} \circ \mathsf{PRODUCT}_{d'} \circ \mathsf{MOD2} \circ \mathsf{AND}_D.$$

Noting the PRODUCTs now take 0/1 inputs, we can replace them with ANDs:

$$\mathsf{SUM}_{n^{d'}} \circ \mathsf{SUM}_{n^{O(d')}} \circ \mathsf{AND}_{d'} \circ \mathsf{MOD2} \circ \mathsf{AND}_D.$$

Taking the Fourier expansion of the AND function (see, e.g., (5) in Lemma 5.4), we can replace each AND gate with a SUM of $2^{d'}$ MOD2s of fan-in $\leq d'$:

$$\mathsf{SUM}_{n^{d'}} \circ \mathsf{SUM}_{n^{O(d')}} \circ \mathsf{SUM}_{2^{d'}} \circ \mathsf{MOD2} \circ \mathsf{AND}_D.$$

Merging the SUMs, our final expression has the form:

$$\mathsf{SUM}_{n^{O(d')}} \circ \mathsf{MOD2} \circ \mathsf{AND}_D.$$

Finally, we want to choose a value of t so that the fan-in of the SUM is subexponential, and the fan-ins of the AND's are sublinear (which will also imply that the fan-in of the MOD2's are subexponential). Let $t = n^{1+\beta}$, where β is an arbitrary number between $\alpha < \beta < \delta/(1-\delta)$. Note that

$$d' = O(\sqrt{k}) = O(\sqrt{s/t}) = O(n^{1-\frac{\beta-\alpha}{2}}) = O(n^{1-\gamma})$$

for every $0 < \gamma < \frac{\beta-\alpha}{2}$. Also, observe that

$$D = O(t^{1-\delta} \cdot \log s) = O(n^{1-(1-\delta)(\delta/(1-\delta)-\beta)} \log n) = O(n^{1-\gamma})$$

for every $0 < \gamma < (1-\delta)(\delta/(1-\delta)-\beta)$.

From the upper bounds on d' and D, we have that F can be computed by

$$\mathsf{SUM}_{2^{n^{1-\gamma}}} \circ \mathsf{MOD2}_{2^{n^{1-\gamma}}} \circ \mathsf{AND}_{n^{1-\gamma}}$$

for some $\gamma > 0$. □

The above formula depth reduction shows that, if there are more efficient probabilistic polynomials for DeMorgan formulas (and we have no reason to doubt this), then super-cubic formulas have interesting representations as approximate sums of sub-exponentially many sub-linear degree $\mathbb{F}_2$-polynomials. Recent work [Wil18, CW19] can already be applied to prove interesting lower bounds against approximate sums of 2^{n^α} $\mathbb{F}_2$-polynomials of degree n^β, where $\alpha + \beta < 1$. The remaining challenge will be to prove lower bounds when $\max\{\alpha, \beta\} < 1$.

4 Circuit Depth Reductions

In this section, we present new depth reductions for circuits with unrestricted depth.

4.1 Linear Circuits

We start by considering *linear* circuits, i.e., circuits consisting of $\oplus$ gates only. For technical reasons, we assume that there are $n+1$ input gates in a linear circuit: $x_1, \ldots, x_n$ as well as the constant 0. For a matrix $M \in \{0,1\}^{m \times n}$, we say that a linear circuit $\mathcal{C}$ with m outputs computes the linear transformation M if the i-th output of $\mathcal{C}(x)$ equals the i-th row of Mx for all $x \in \{0,1\}^n$, treating $\mathcal{C}(x)$ as the vector of output values. We say that a linear circuit $\mathcal{C}$ computing M is *optimal* if no circuit of smaller size computes M.

The main result of this subsection asserts that matrices computable by small linear circuits are not too rigid. The contrapositive says: to get an improved lower bound on the size of linear circuits, it suffices to construct a matrix with good rigidity parameters. Below, we restate the corresponding theorem formally and then prove it.

Theorem 1.2. *For every matrix $M \in \mathbb{F}_2^{m \times n}$ of linear circuit complexity s, $\mathbb{R}_M(\lfloor s/4 \rfloor) \le 16$.*

Proof. Let $\mathcal{C}$ be an optimal circuit of size s computing M. If $s < 16$ or the depth of $\mathcal{C}$ is at most 4, then each output depends on at most 16 variables. Hence M is 16-sparse and the theorem statement holds. Consider this as the base case of an induction on s.

For the induction step, we "normalize" $\mathcal{C}$. Namely, we show how to express M as the (modulo 2) sum of two $\mathbb{F}_2$-matrices A and B, where A is 16-sparse (each row has ≤ 16 ones) and B has rank at most $\lfloor s/4 \rfloor$. Note that if $\mathcal{C}$ has an output gate H of depth at most 4, then H depends on at most $2^4 = 16$ inputs. Thus the corresponding row r_H of M has at most 16 ones. Consider the $(m-1) \times n$ matrix M_{-H} obtained by removing r_H from M. We claim that $\mathbb{R}_{M_{-H}}(\lfloor s/4 \rfloor) \le 16$ implies $\mathbb{R}_M(\lfloor s/4 \rfloor) \le 16$. Indeed, suppose $M_{-H} = A_{-H} \oplus B_{-H}$ where A_{-H} is 16-sparse and $\operatorname{rank}(B_{-H}) \le \lfloor s/4 \rfloor$. To get matrices A and B for M, we simply add the row r_H to A_{-H} and a corresponding all-zero row to B_{-H}. Clearly, the resulting matrix A is 16-sparse and the rank of the resulting matrix B does not change. Thus, in the following, we assume WLOG that $\mathcal{C}$ has no output gates of depth at most 4. Our crucial step is the following claim.

Claim 4.1. *Let $\mathcal{C}$ be an optimal linear circuit computing $M \in \{0,1\}^{m \times n}$ such that $s(\mathcal{C}) \ge 16$, and no output gate of $\mathcal{C}$ has depth smaller than 5. Then there is a gate G in $\mathcal{C}$ and a linear circuit $\mathcal{C}'$ computing a matrix $M' \in \{0,1\}^{m \times n}$ with the properties:*

1. *$s(\mathcal{C}') \le s(\mathcal{C}) - 4$, and*
2. *for every $x \in \{0,1\}^n$, if $G(x) = 0$ then $\mathcal{C}(x) = \mathcal{C}'(x)$.*

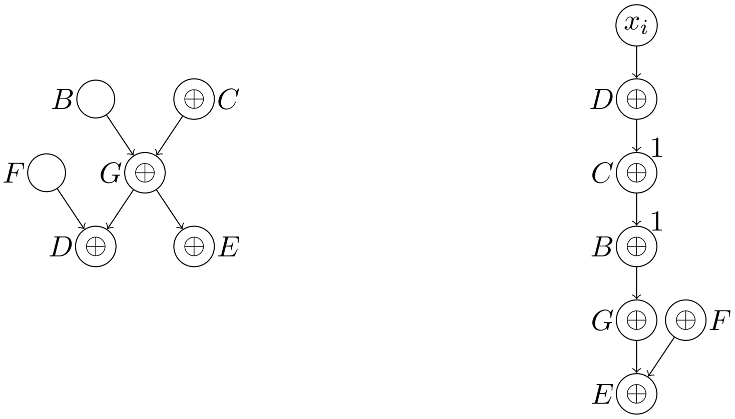
For now, suppose the claim is proved. Consider the circuit $\mathcal{C}'$, gate G in $\mathcal{C}$, and matrix M' provided by Claim 4.1. Let $g \in \{0,1\}^{1 \times n}$ be the characteristic vector of the linear function computed by G, so that $G(x) = gx$. By the claim, $gx = 0$ implies $(M \oplus M')x = 0$. Hence $(M \oplus M')$ is either the zero matrix, or it defines the same linear subspace as g: $M \oplus M' = tg$ for a vector $t \in \{0,1\}^{m \times 1}$.

By the induction hypothesis, $M' = A' \oplus B'$ where A' is 16-sparse, and $\operatorname{rank}(B') \le \lfloor \frac{s-4}{4} \rfloor = \lfloor \frac{s}{4} \rfloor - 1$. Thus, $M = A' \oplus B$, where the matrix $B = B' \oplus tg$ has rank at most $\lfloor s/4 \rfloor$ by subadditivity of the rank function. □

We now turn to proving the remaining claim.

Proof of Claim 4.1.

Case 1: There is a gate G in $\mathcal{C}$ of depth at least 2 and at most 4, and has out-degree at least 2. Let the predecessors of G be B and C, and call two of its successors D and E, see Figure 4.1 (in this and the following figures, we write the out-degrees of some of the gates near them). The circuit $\mathcal{C}'$ is obtained from $\mathcal{C}$ by "assigning" the output of G to be 0. Note that $B(x) = C(x)$ for all $x \in \{0,1\}^n$ where $G(x) = 0$. At least one of B and C must be an internal gate (otherwise G would have depth 1), let it be C. Since C computes the same function as B, it may be removed from $\mathcal{C}'$: we remove it, and replace every wire of the form $C \to H$ by a new wire $B \to H$. Note that neither G nor C is an output gate. Now, we show that both D and E can also be removed. Let us focus on the gate D (for E it is shown similarly) and call its other predecessor F. Since $G = 0$, the gate D computes the same function as F. This means that one may remove D: we remove it and replace every wire $D \to H$ by a wire $F \to H$. If D happens to be an output gate, we move the corresponding output label from D to F.



Case 1: assuming $G = 0$, the gate G is removed, B is replaced by C, and D and E are replaced by their other predecessors.

Case 2: assuming $G = 0$, the gates B, C, and G are removed whereas E is replaced by F.

Figure 1: Cases in the proof of Claim 4.1.

Case 2: All gates of depth at least 2 and at most 4 have out-degree exactly 1 in $\mathcal{C}$. Take a gate G of depth 4 and trace back its longest path to an input: $x_i \to D \to C \to B \to G$. Let also E be the successor of G (which exists because C has depth at least 5). By assumption, gates B and C have out-degree 1. This means that in $\mathcal{C}$ they are only used for computing the gate G. This, in turn, means that assuming $G = 0$, we can remove G, B, and C (note none of them is an output). Finally, the gate E can be replaced by the other input F of E (note $F \notin \{B, C, G\}$, since $\mathcal{C}$ is optimal).

This completes the proof. □

Remark 4.2. *Extending the same ideas, one can show that any linear circuit $\mathcal{C}$ of size s can be computed by an $OR_{2^{\lceil \frac{s}{4} \rceil}} \circ AND_{s \cdot 2^{14}} \circ OR_{16}$ circuit. For this, one considers two optimal circuits $\mathcal{C}_0$*

and $\mathcal{C}_1$ resulting from $\mathcal{C}$ by assuming $G = 0$ and $G = 1$, respectively. As shown in the proof, both $\mathcal{C}_0$ and $\mathcal{C}_1$ have size at most $s - 4$. One then proceeds by induction. We illustrate this approach in full detail in the next subsection.

Remark 4.3. *The proof of Theorem 1.2 gives a decomposition $M = A \oplus B = A \oplus (C \cdot D)$, where $A \in \mathbb{F}^{m \times n}$ is 16-sparse, $C \in \mathbb{F}^{m \times s/4}$ is composed of vectors t, and $D \in \mathbb{F}^{s/4 \times n}$ is composed of vectors g. Since the chosen gate G always has depth at most four, the vector g is 16-sparse. Thus, we in fact have a decomposition $M = A \oplus (C \cdot D)$, where both A and D are 16-sparse. In particular, the row-space of M is spanned by the union of row-spaces of A and D. This implies that the row-space of M can be spanned by at most $(m + \frac{s}{4})$ 16-sparse vectors. The corresponding matrix property is called* outer dimension, *and it is studied in [PP06, Lok09]. While the current lower bounds on the outer dimension of explicit matrices do not lead to new circuit lower bounds, it would be interesting to study their applications in this context.*

4.2 General Boolean Circuits

In this section, we study the following natural question: given a Boolean circuit[6] and given an integer $k \geq 2$, what is the smallest $\mathrm{OR} \circ \mathrm{AND} \circ \mathrm{OR}_k$ circuit computing the same function? To this end, we introduce the following notation. For an integer $k \geq 2$, we define $\alpha(k)$ as the infimum of all values α such that any circuit of size s can be rewritten as a $\mathrm{OR}_{2^{\alpha s}} \circ \mathrm{AND} \circ \mathrm{OR}_k$ circuit.

For proving upper bounds on $\alpha(k)$ it will be convenient to consider the following class of circuits. Let $\mathrm{OR}_p \circ \mathrm{AND}_q \circ C(r)$ be a class of circuits with an output OR that is fed by at most p AND's of at most q circuits of size at most r.

Theorem 4.4. *Every circuit of size s can be computed as:*

1. *an $OR_{2^{\lceil \frac{s}{2} \rceil}} \circ AND_{\lceil \frac{s}{2} \rceil} \circ C(1)$ circuit;*
2. *an $OR_{2^{\lceil \frac{s}{3.9} \rceil}} \circ AND_{\lceil \frac{s}{3} \rceil} \circ C(15)$ circuit.*

Note that any circuit of size r depends on at most $r + 1$ variables, and hence can be written as an $(r + 1)$-CNF with at most 2^r clauses. Therefore every $\mathrm{OR}_p \circ \mathrm{AND}_q \circ C(r)$ circuit can be easily converted into a $\mathrm{OR}_p \circ \mathrm{AND}_{q2^r} \circ \mathrm{OR}_{r+1}$ circuit. Thorem 1.1, which we restate below, is then an immediate corollary of Theorem 4.4. In turn, it implies that $\alpha(2) \leq \frac{1}{2}$ and $\alpha(16) \leq \frac{1}{3.9}$.

Theorem 1.1. *Every circuit of size s can be computed as an $OR_{2^{\lceil \frac{s}{2} \rceil}} \circ AND_s \circ OR_2$ circuit and as an $OR_{2^{\lceil \frac{s}{3.9} \rceil}} \circ AND_{2^{14} \cdot s} \circ OR_{16}$ circuit.*

Proof of Theorem 4.4. Both parts are proven in a similar fashion. We proceed by induction on s. The base case is when s is small. We then just have an $\mathrm{OR}_1 \circ \mathrm{AND}_1 \circ C(s)$ circuit.

For the induction step we take a gate G of $\mathcal{C}$ and consider two circuits $\mathcal{C}_0$ and $\mathcal{C}_1$ where $\mathcal{C}_i$ computes the same as $\mathcal{C}$ on all inputs $\{x \in \{0,1\}^n \colon G(x) = i\}$. We may assume both $\mathcal{C}_i$'s are minimal size among all such circuits. Since $\mathcal{C}_i$ can be obtained from $\mathcal{C}$ by removing the gate G (as it computes the constant i on the corresponding subset of the Boolean hypercube), we conclude that $s(\mathcal{C}_i) < s$. This allows us to proceed by induction. Assume that by the induction hypothesis $\mathcal{C}_i$ is guaranteed

[6] In this section we consider functions with one output, but these results can be trivially generalized to the multi-output case.

to be expressible as an $\mathrm{OR}_{p_i} \circ \mathrm{AND}_{q_i} \circ C(r_i)$ circuit. We use the following identity to convert $\mathcal{C}$ into the required circuit:

$$\mathcal{C}(x) \equiv ([G(x) = 0] \wedge \mathcal{C}_0(x)) \vee ([G(x) = 1] \wedge \mathcal{C}_1(x))\,. \tag{1}$$

Assume that the subcircuit of $\mathcal{C}$ computing the gate G has at most t gates. We claim that $[G(x) = i] \wedge \mathcal{C}_i$ can be written as an $\mathrm{OR}_{p_i} \circ \mathrm{AND}_{q_i+1} \circ C(\max\{r_i, t\})$ circuit. For this, we just feed a new circuit computing G to every AND gate. Plugging this into (1), gives an

$$\mathrm{OR}_{p_0+p_1} \circ \mathrm{AND}_{\max\{q_0,q_1\}+1} \circ C(\max\{t, r_0, r_1\}) \tag{2}$$

circuit for computing $\mathcal{C}$.

Below, we provide details specific to each of the two items from the theorem statement. In particular, we estimate the parameters p_i's, q_i's, r_i's, and t and plug them into (2).

1. The base case is $s = 1$. Then $\mathcal{C}$ consists of a single gate and can be expressed as an $\mathrm{OR}_1 \circ \mathrm{AND}_1 \circ C(1)$ circuit. For the induction step, assume that $s \geq 2$ and take a gate A that depends on two variables. Let $G = A$, hence $t = 1$. The gate A must have at least one successor (otherwise $\mathcal{C}$ can be replaced by a circuit with smaller than s gates). Clearly, A and its successors are not needed in $\mathcal{C}_i$'s. Hence, by the induction hypothesis $p_i \leq 2^{\frac{s-2}{2}+1}$, $q_i \leq \frac{s-2}{2} + 1$, $r_i \leq 1$. Plugging this into (2) gives the desired result.

2. Take a gate A that is fed by two variables x and z and has the maximum distance to an output. If its distance to output is at most 4, then $s(\mathcal{C}) \leq 15$ and we just rewrite it as an $\mathrm{OR}_1 \circ \mathrm{AND}_1 \circ C(15)$ circuit. This is the base case. Assume now that the distance from A to the output gate is at least 5. In the analysis below, we always "follow" the longest path from A to the output. This allows us to conclude that any such path is long enough and hence each gate considered has positive out-degree (i.e., is not an output). Moreover, each gate on this path cannot depend on too many variables. Let B be a successor of A on the longest path to the output.

   In the five cases below, we show that we can always find a gate G that $s(G) \leq 15$ and both $s(\mathcal{C}_0)$ and $s(\mathcal{C}_1)$ are small enough. In particular, $s(\mathcal{C}_0), s(\mathcal{C}_1) \leq s - 4$ works for us: $p_0 + p_1 \leq 2 \cdot 2^{\lceil \frac{s-4}{3.9} \rceil} < 2^{\lceil \frac{s}{3.9} \rceil}$, $\max\{q_0, q_1\} + 1 \leq \lceil \frac{s-4}{3} \rceil + 1 < \lceil \frac{s}{3} \rceil$.

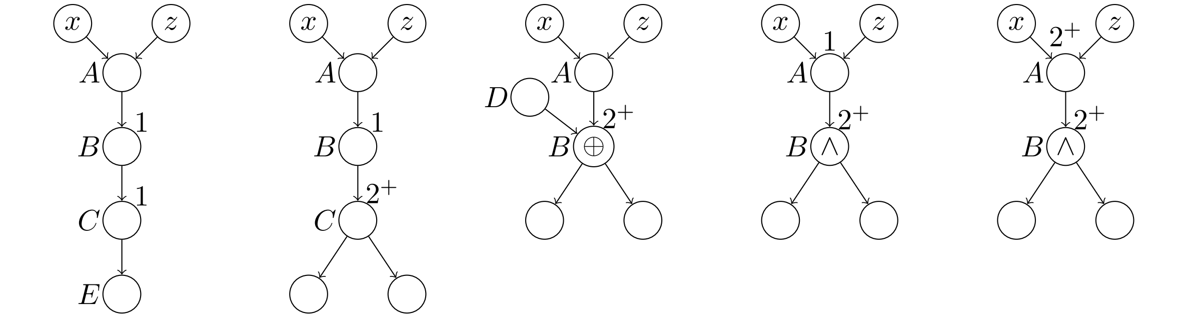
   See Figure 2 for an illustration of the five cases. For a gate G, by $\mathrm{out}(G)$ we denote the out-degree of G.

   Case 1: $\mathrm{out}(B) = 1$. Let C be the successor of B.

   Case 1.1: $\mathrm{out}(C) = 1$. Let E be the successor of C. Let $G = E$. In $\mathcal{C}_i$'s, one removes B, C (as they were only needed to compute E that is now a constant), E, and the successors of E.

   Case 1.2: $\mathrm{out}(C) \geq 2$. Let $G = C$. In $\mathcal{C}_i$'s, one removes B, C, and the successors of C.

   Case 2: $\mathrm{out}(B) \geq 2$. Let D be the other input of B. It may be a gate or an input variable. If B computes a constant Boolean binary operation or an operation that depends on A or D only, then $\mathcal{C}$ is not optimal. Otherwise, B computes one of the following two types of functions (either linear or quadratic polynomial over $\mathbb{F}_2$):



Case 1.1: when E is constant, one removes B, C, E, and successors of E.

Case 1.2: when C is constant, one removes B, C, and successors of C.

Case 2.1: when B is constant, one removes B and its successors, replace A by $D \oplus c$.

Case 2.2.1: when B is constant, one removes B and its successors, and A.

Case 2.2.2: when B is constant, one removes B and its successors; moreover, $B = 1$ it forces A to be a constant and removes A and its successors.

Figure 2: Cases in the proof of the second part of Theorem 4.4.

Case 2.1: $B(A, D) = A \oplus D \oplus c$ where $c \in \{0, 1\}$. Let $G = B$. In $\mathcal{C}_i$'s, one immediately removes B and its successors. Also, in $\mathcal{C}_i$, $D \oplus A = i \oplus c$. Hence, A may be replaced by $D \oplus i \oplus c$.

Case 2.2: $B(A, D) = (A \oplus a) \cdot (D \oplus d) \oplus c$ where $a, d, c \in \{0, 1\}$.

Case 2.2.1: $\operatorname{out}(A) = 1$. Let $G = C$. In $\mathcal{C}_i$'s, one removes B, its successors, and A.

Case 2.2.2: $\operatorname{out}(A) \geq 2$. Let D be the other successor of B. Let $G = B$. In $\mathcal{C}_i$'s, one removes B and its successors. Also, $B = c \oplus 1$ forces $A = a \oplus 1$ and $D = d \oplus 1$. Hence, in $\mathcal{C}_{c \oplus 1}$ two additional gates are removed: A and its successors (if a successor of B happens to be a successor of A also, then it is a function on A and D and the circuit can be simplified, which contradicts its optimality). Hence, $p_0 + p_1 \leq 2^{\lceil \frac{s-3}{3.9} \rceil} + 2^{\lceil \frac{s-5}{3.9} \rceil}$. This is smaller than $2^{\lceil \frac{s}{3.9} \rceil}$ since $2^{-\frac{3}{3.9}} + 2^{-\frac{5}{3.9}} < 1$.

This completes the proof. □

Remark 4.5. *It is not difficult to see that the output OR gate is a "disjoint OR", and can be replaced by a SUM gate over the integers. In other words, for every $x \in \{0, 1\}^n$, at most one subcircuit feeding into the OR gate may evaluate to 1. This holds because we always consider two mutually exclusive cases: $G = 0$ or $G = 1$.*

4.3 Properties of $\alpha(k)$

We start by observing a lower bound on $\alpha(k)$.

Lemma 4.6. *For any integer $k \geq 2$, $\alpha(k) \geq 1/k$.*

Proof. Let $\oplus_n$ denote the parity function of n inputs. It has 2^{n-1} inputs where it is equal to 1 and all these inputs are isolated, that is, the Hamming distance between any pair of them is at least 2. As proven by Paturi, Pudlák, and Zane [PPZ97], every k-CNF has at most $2^{n(1-1/k)}$ isolated satisfying assignments. This implies that $\oplus_n$ cannot be computed by an OR of fewer than $2^{n/k-1}$ k-CNFs. Since $s(\oplus_n) = n - 1$, this implies that

$$\alpha(k) \geq \frac{\frac{n}{k} - 1}{n - 1} .$$

Since this must hold for arbitrary large n, $\alpha(k) \geq 1/k$. □

Thus, we know the exact value of $\alpha(2) = \frac{1}{2}$. This immediately implies a circuit lower bound of $2n - o(n)$ for BCH codes. Indeed, it was shown in [PSZ97] that when the bottom fan-in is restricted to $k = 2$, then BCH codes require depth-3 circuits of size $2^{n-o(n)}$. And, since $\alpha(2) = \frac{1}{2}$, they must have circuit complexity at least $2n - o(n)$.

One can use techniques from Theorem 4.4 to prove an upper bound of $\alpha(3) \leq \frac{\log_2 3}{4}$. Thus, we know that

$$\frac{1}{3} \leq \alpha(3) \leq \frac{\log_2 3}{4} < 0.3963 .$$

We conjecture that the upper bound on α_3 is tight. One way to prove this would be to find the s_3^3 complexity of the inner product function: $\mathrm{IP}(x_1, \ldots, x_n) = x_1x_2 \oplus x_3x_4 \oplus \cdots \oplus x_{n-1}x_n$. In particular, if the upper bound shown in the next lemma is tight, then $\alpha(3) = \frac{\log_2 3}{4}$.

Lemma 4.7.

1. $2^{\frac{n}{4}} \leq s_3^2(\mathrm{IP}) \leq 2^{\frac{n}{2}-o(n)}$.
2. $2^{\frac{n}{6}} \leq s_3^3(\mathrm{IP}) \leq 3^{\frac{n}{4}}$.

Proof. Note that by substituting every other input of IP by 1, one gets the parity function $\oplus_{\frac{n}{2}}$ on the remaining $n/2$ inputs. Now both lower bounds follow from the corresponding lower bounds for the parity function: $s_3^2(\oplus_k) \geq 2^{\frac{k}{2}}$ and $s_3^3(\oplus_k) \geq 2^{\frac{k}{3}}$.

1. The first upper bound follows from the fact that $\mathrm{IP}(x_1, \ldots, x_n) = 1$ iff there is an odd number of ones among

   $$p_1 = x_1x_2,\, p_2 = x_3x_4,\, \ldots, p_{\frac{n}{2}} = x_{n-1}x_n .$$

   Hence,

   $$\mathrm{IP}(x_1, \ldots, x_n) \equiv \bigvee_{S \subseteq [\frac{n}{2}] \colon |S| \bmod 2 = 1} \left(\bigwedge_{i \in S} [p_i = 1] \wedge \bigwedge_{i \notin S} [p_i = 0] \right) .$$

   It remains to note that each $[p_i = c]$ can be expressed as a 2-CNF because p_i depends on two variables.

2. For the second upper bound, note that $\mathrm{IP}(x_1, \ldots, x_n) = 1$ iff there is an odd number of 1's among

   $$p_1 = x_1x_2 \oplus x_3x_4,\, p_2 = x_5x_6 \oplus x_7x_8,\, \ldots, p_{\frac{n}{4}} = x_{n-3}x_{n-2} \oplus x_{n-1}x_n .$$

To compute IP by a depth 3 circuit, we go through all possible $2^{\frac{n}{4}-1}$ values of $p_1, \dots, p_{\frac{n}{4}}$ such that an odd number of them is equal to 1:

$$\mathrm{IP}(x_1, \dots, x_n) \equiv \bigvee_{S \subseteq [\frac{n}{4}] \colon |S| \bmod 2 = 1} \left(\bigwedge_{i \in S} [p_i = 1] \wedge \bigwedge_{i \notin S} [p_i = 0] \right) \tag{3}$$

Now, we show that $[p_i = 0]$ can be written as a single 3-CNF, whereas $[p_i = 1]$ can be expressed as an OR of two 3-CNFs. W.l.o.g. assume that $i = 1$. The clauses of a 3-CNF expressing $[p_i = 0]$ should reject all assignments to $x_1, x_2, x_3, x_4 \in \{0, 1\}$ where $\mathrm{IP}(x_1, x_2, x_3, x_4) = 1$. In all such assignments, one of the two monomials (x_1x_2 and x_3x_4) is equal to 0 whereas the other one is equal to 1. Hence, one needs to write down a set of clauses rejecting the following four partial assignments: $\{x_1 = 0, x_3 = x_4 = 1\}$, $\{x_2 = 0, x_3 = x_4 = 1\}$, $\{x_1 = x_2 = 1, x_3 = 0\}$, $\{x_1 = x_2 = 1, x_4 = 0\}$. Thus,

$$[p_1(x_1, x_2, x_3, x_4) = 0] \equiv (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \,.$$

In turn, to express $[p_1 = 1]$ as an OR of two 3-CNFs we consider both assignments to x_1:

$$[p_1(x_1, x_2, x_3, x_4) = 1] \equiv ((x_1) \wedge [x_2 \oplus x_3x_4 = 0]) \vee ((\neg x_1) \wedge [x_3x_4 = 1]) \,.$$

It remains to note that each of $[x_2 \oplus x_3x_4 = 0]$ and $[x_3x_4 = 1]$ can be written as a 3-CNF. Let $[p_i = 0] \equiv P_i$ and $[p_i = 1] \equiv ((x_i) \wedge Q_i) \vee ((\neg x_i) \wedge R_i)$ where P_i, Q_i, and R_i are 3-CNFs. One may then expand (3) as follows:

$$\bigvee_{S \subseteq [\frac{n}{4}] \colon |S| \bmod 2 = 1} \left(\bigvee_{T \subseteq S} \left(\bigwedge_{i \in T} ((x_i) \wedge Q_i) \wedge \bigwedge_{i \in S \setminus T} ((\neg x_i) \wedge R_i) \wedge \bigwedge_{i \notin S} P_i \right) \right)$$

The fan-in of the resulting OR-gate is

$$\sum_{S \subseteq [\frac{n}{4}] \colon |S| \bmod 2 = 1} 2^{|S|} \leq \sum_{i=0}^{\frac{n}{4}} \binom{n/4}{i} 2^i = 3^{\frac{n}{4}} \,.$$

□

Open Problem 4.1. *Determine* $s_3^3(\mathrm{IP})$.

Besides finding the exact values of $\alpha(k)$, it would be interesting to find out whether every circuit of *linear size* can be computed by a non-trivial depth 3 circuit with constant bottom fan-in. We restate this open problem below.

Open Problem 1.1. *Prove or disprove: for any constant c, any circuit of size cn can be computed as an*

$$OR_{2^{(1-\delta(c))n}} \circ AND \circ OR_{\gamma(c)}$$

circuit, for some $\delta(c) > 0$ *and integer* $\gamma(c) \geq 1$.

This paper supports the conjecture by showing that it holds for small values of c. As another example, we can consider a class of functions where we know linear *upper bounds* on circuit complexity. For any *symmetric* function f (i.e., a function whose value depends only on the sum over integers of the input bits) we know that $s(f) \leq 4.5n + o(n)$ [DKKY10]. It is also known [PSZ97, Wol06] that symmetric functions can be computed by relatively small depth-3 circuits: $s_3^k(f) \leq \operatorname{poly}(n) \cdot (1 + 1/k)^n$ (and this bound is tight [Wol06]).

Since in our depth reduction results, we always get k-CNFs with small linear number of clauses, it is interesting to study the expressiveness of OR of exponential number of such k-CNFs. Let us define $\alpha(k, c)$ as the infimum of all values α such that any circuit of size at most cn can be computed as an $\mathrm{OR}_{2^{\alpha n}} \circ \mathrm{AND}_{cn} \circ \mathrm{OR}_k$. We can upper bound the rate of convergence of $\alpha(k, c)$ using the following width reduction result for CNF-formulas [Sch05, CIP06].

Theorem 4.8 ([Sch05, CIP06]). *For any constant $0 < \varepsilon \leq 1$ and a function $C \colon \mathbb{N} \to \mathbb{N}$, any CNF formula f with n variables and $n \cdot C(n)$ clauses can be expressed as $f = OR_{i=1}^t f_i$, where $t \leq 2^{\varepsilon n}$ and each f_i is a k-CNF formula with at most $n \cdot C(n)$ clauses, where $k = O\left(\frac{1}{\varepsilon} \cdot \log\left(\frac{C(n)}{\varepsilon}\right)\right)$.*

For our applications, we are interested in $\alpha(k, c)$ for small fixed c. Since for every c, $\alpha(k, c)$ is a non-increasing bounded sequence, we let $\alpha(\infty, c) = \lim_{k \to \infty} \alpha(k, c)$. Then Theorem 4.8 implies that $\alpha(k, c) \geq \alpha(\infty, c) \geq \alpha(k, c) - O\big(\frac{\log(ck)}{k}\big)$.

5 Applications

In this section, we state formally the results that are presented in the last three row-blocks of Table 1. Namely, we show that improving the parameters for the known explicit constructions of the following pseudorandom objects imply circuits lower bounds via depth reduction techniques presented in the previous section:

- functions that are not constant on any large algebraic variety in $\{0, 1\}^n$ defined by polynomials of small degree (such functions are called dispersers);
- functions that agree with any polynomial of small degree on roughly half of the points in $\{0, 1\}^n$;
- matrices that are far from matrices of small rank.

For comparison, we also show what these tools give when applied to Valiant's reductions.

5.1 Dispersers

In this section we show that dispersers for algebraic varieties over $\mathbb{F}_2$ cannot be computed by small circuits. We note that dispersers for varieties of degree one have been used for proving lower bounds on unrestricted circuits [DK11, FGHK16], and it is known that an explicit construction of a disperser for varieties of degree two would slightly improve the known circuit lower bounds [GK16]. Now we show that dispersers for varieties of degree 16 will give new circuit lower bounds via a new simple method.

Definition 5.1. *A set $S \subseteq \{0,1\}^n$ is called an (d,m)-*variety *if it is a set of common roots of at most m polynomials of degree at most d:*

$$S = \{x \in \{0,1\}^n \colon p_1(x) = \cdots = p_m(x) = 0,\, \deg(p_i) \le d \text{ for all } 1 \le i \le m\}\,.$$

*A set S is called a d-*variety *(or a variety of degree d) if it is an (d,∞)-variety.*

Definition 5.2. *A Boolean function $f \colon \{0,1\}^n \to \{0,1\}$ is called a (d,m,s)-*disperser *(for parameters d, m, and s which possibly depend on n) if f is non-constant on any (d,m)-variety $S \subseteq \{0,1\}^n$ of size larger than s.*

We will make use of the Sparsification Lemma first proven by Impagliazzo, Paturi and Zane [IPZ01]. The dependence of C on k was later improved in [CIP06]. (And this is essentially tight by [MRW05].)

Theorem 5.1 (Corollary 1 in [IPZ01], Section 6 in [CIP06])**.** *For all $\varepsilon > 0$ and positive k, there exists C such that any k-CNF formula f with n variables can be expressed as $f = OR_{i=1}^t f_i$, where $t \le 2^{\varepsilon n}$ and each f_i is a k-CNF formula with at most Cn clauses, where $C = O\left(\left(\frac{k}{\varepsilon}\right)^{3k}\right)$.*

Now we are ready to state the main result of this section.

Theorem 5.2. *Let $f \colon \mathbb{F}^n \to \mathbb{F}$ be a function with $|f^{-1}(1)| \ge |f^{-1}(0)|$ and $\varepsilon > 0$ be a constant.*[7]

- *If f is an $(16, 1.3(1-\varepsilon)n, 2^{\varepsilon n})$-disperser, then $s(f) \ge 3.9(1-\varepsilon)n - 4$.*
- *If f is an $(\omega(1), O(n), 2^{(1-\varepsilon)n})$-disperser, then $s_{sp}(f) = \omega(n)$.*
- *If f is $(2^{(\log n)^{1-o(1)}}, \infty, 2^{(1-\varepsilon)n})$-disperser, then $s_{\log}(f) = \omega(n)$.*
- *If f is $(n^{\varepsilon}, \infty, 2^{n-\omega(n/\log\log n)})$-disperser, then $s_{\log}(f) = \omega(n)$.*

Proof.

- From Theorem 4.4, we know that if f is computable by a circuit of size s, then f is also computable by a circuit $\mathcal{C} \in \mathrm{OR}_{2^{s/3.9}} \circ \mathrm{AND}_{s/3} \circ C(15)$. Let $t = 2^{s/3.9}$, and let $f_1, \ldots, f_t \colon \mathbb{F}^n \to \mathbb{F}$ be the t functions computed in the gates of the AND level of $\mathcal{C}$. Since $f = \mathrm{OR}_{i=1}^t f_i$, we have that $f^{-1}(1) = \bigcup_{i=1}^t f_i^{-1}(1)$. Thus,

$$2^{n-1} \le \left|f^{-1}(1)\right| \le \sum_{i=1}^{t} |f_i^{-1}(1)| \le t \cdot \max_i |f_i^{-1}(1)|\,. \tag{4}$$

  Each f_i is an $\mathrm{AND}_{s/3} \circ C(15)$, that is, a set of common roots of $s/3$ polynomials of degree 16 (recall that over $\mathbb{F}_2$ every monomial is multilinear; hence a circuit of size 15 computes a polynomial of degree at most 16). Since f is a disperser for varieties of size $2^{\varepsilon n}$ defined by $s/3$ polynomials of degree 16, each $f_i^{-1}(1) \le 2^{\varepsilon n}$. Now, (4) implies that $s/3.9 \ge n - \varepsilon n - 1$.

[7] If $|f^{-1}(1)| < |f^{-1}(0)|$, one can consider the negation of f, since taking negations does not change the disperser parameters.

- The proofs of items (2)–(4) of this theorem follow the same pattern, so we only present the proof of the second item. Assume, towards a contradiction, that an $(\omega(1), O(n), 2^{(1-\varepsilon)n})$-disperser f can be computed by a series-parallel circuit of size cn. From Theorem 2.1, such a circuit can be expressed as a circuit $\mathcal{C} \in \mathrm{OR}_{2^{\frac{\varepsilon n}{3}}} \circ \mathrm{AND} \circ \mathrm{OR}_k$ for $k = k(c, \varepsilon)$. By Theorem 5.1, each k-CNF computed by the AND gates of $\mathcal{C}$, can be replaced by an OR of $2^{\frac{\varepsilon n}{3}}$ k-CNFs with Cn clauses each where $C = C(\delta, \varepsilon)$. Let $t = 2^{\frac{2\varepsilon n}{3}}$, and let $f_1, \ldots, f_t \colon \mathbb{F}^n \to \mathbb{F}$ be the t k-CNFs with Cn clauses whose OR computes f. Now we have that each f_i is an $\mathrm{AND}_{Cn} \circ \mathrm{OR}_k$, that is, a set of common roots of Cn polynomials of degree k (each computing an OR_k). From the disperser property of f, we have that each f_i computes at most $2^{(1-\varepsilon)n}$ ones of f. Therefore, in order to compute all $\geq 2^{n-1}$ ones of f, t must be greater than $2^{\varepsilon n - 1}$, which contradicts the definition $t = 2^{\frac{2\varepsilon n}{3}}$.

□

We remark that in the first item of Theorem 5.2, even dispersers for varieties defined by $1.3(1-\varepsilon)n$ functions of 16 variables (rather than all polynomials of degree 16) will suffice for proving a lower bound.

In order to prove a new circuit lower bound against unrestricted circuits, it suffices to construct a $(16, 1.05n, 2^{0.2n})$-disperser. There are known constructions of dispersers for constant-degree varieties over large fields [Dvi12, BSG12, LZ19]. For $\mathbb{F}_2$, a long line of work achieved almost optimal dispersers for degree $d = 1$ varieties, which are not constant on sets of size $2^{(\log n)^c}$ for a constant c [Li16]. Also, the known constructions can handle large varieties of large degrees [Rem16], or smaller varieties of size $2^{\alpha n}$ of constant degree (for a constant α) [LZ19]. On the other hand, the result of Cohen and Tal [CT15, Theorem 5], together with an efficient construction of affine dispersers from [Li16], gives an explicit construction of $\left(16, \frac{n}{(\log n)^c}, 2^{o(n)}\right)$-disperser (it handles varieties of the desired size, but only defined by fewer polynomials). Thus, although the currently known constructions do not suffice for proving new lower bounds, they are tantalizingly close to the ones needed for a simple proof of circuit lower bounds via Theorem 4.4.

We conclude this section with a simple counting argument showing that a random function is a disperser with great parameters.

Lemma 5.3. *Let $d = d(n)$, $m = m(n)$, $s = s(n)$ be such that $s > 3dmn^d$. Then a random function $f \colon \{0,1\}^n \to \{0,1\}$ is a (d, m, s)-disperser with probability $1 - o(1)$.*

Proof. Consider a function f that is not a (d, m, s)-disperser. That is, f is constant on some (d, m)-variety. In particular, f can be uniquely specified by

1. a (d, m)-variety V where f is constant,
2. one of the two possible constant values that f takes on V,
3. values at the remaining (at most $2^n - s$) points.

There are $k = \sum_{i=0}^{d} \binom{n}{i} \leq 2dn^d$ monomials of degree at most d over $\{x_1, \ldots, x_n\}$ (as any monomial is multilinear). Therefore, there are 2^k polynomials of degree at most d, and at most 2^{mk} (d, m)-varieties. Therefore, the number of functions f which are not (d, m, s)-dispersers is bounded from above by

$$2^{mk} \cdot 2 \cdot 2^{2^n - s} \leq 2^{2dn^d m + 1 + 2^n - s} \leq 2^{2^n} \cdot o(1)$$

Thus, a random function is an (d, k, s)-disperser with probability at least $1 - o(1)$. □

5.2 Correlation with Polynomials

In this section we show that a function that has small correlation with low-degree polynomials has high circuit complexity. We show this by using a known connection between correlation with polynomials and dispersers for varieties.

Definition 5.3. *For two functions $f, g\colon \mathbb{F}^n \to \mathbb{F}$, we define their correlation as*

$$\mathrm{Cor}(f,g) = \left|\Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)]\right| ,$$

where x is drawn uniformly at random from $\mathbb{F}^n$.

By $\mathrm{Cor}(f,d)$ we denote the correlation of a function f with polynomials of degree d:

$$\mathrm{Cor}(f,d) = \max_g \mathrm{Cor}(f,g) ,$$

where the maximum it taken over all polynomials g of degree at most d.

There are several constructions of functions that have small correlation with polynomials of low degree [Raz87, Smo87, BNS92, VW08, Dvi12, Rem16], or sparse polynomials [Vio07]. In particular, the generalized inner product function has correlation $2^{-\Omega\left(\frac{n}{4^d \cdot d}\right)}$ with polynomials of degree d [BNS92], and Viola and Wigderson [VW08] constructed a function with correlation $2^{-\Omega\left(\frac{n}{2^d}\right)}$ with polynomials of degree d. See [Vio09] for an overview of the known bounds on correlation.

We use the fact that small correlation with polynomials of degree d implies small correlation with products of polynomials of degree d, and, as a consequence, a disperser for varieties of degree d.

Lemma 5.4 (Implicit in [Dvi12, CT18, LZ19])**.** *If $\mathrm{Cor}(f,d) \leq \varepsilon$, then f is $(d, \infty, \varepsilon \cdot 2^n)$-disperser.*

Proof. Consider a variety $V = \{x \in \{0,1\}^n \colon q_1(x) = \cdots = q_k(x) = 0\}$, where each $q_i \colon \mathbb{F}^n \to \mathbb{F}$ is a non-constant polynomial of degree at most d. Let $g(x) = \prod_{i=1}^k (q_i(x) \oplus 1)$ be the indicator function of V, and from the Fourier expansion we have

$$g(x) = \frac{\sum_{S \subseteq \{1,\ldots,k\}} (-1)^{\sum_{i \in S} q_i(x)}}{2^k} . \tag{5}$$

Now note that for any $S \subseteq \{1, \ldots, k\}$,

$$\left|\mathbb{E}_x\left[(-1)^{f(x) + \sum_{i\in S} q_i(x)}\right]\right| = \mathrm{Cor}\left(f, \sum_{i \in S} q_i(x)\right) \leq \varepsilon ,$$

because $\sum_{i\in S} q_i(x)$ is a polynomial of degree at most d and $\mathrm{Cor}(f,d) \leq \varepsilon$. Now

$$\begin{aligned}
\left|\mathbb{E}_x\left[(-1)^{f(x)} \cdot g(x)\right]\right| &= \left|\mathbb{E}_x\left[(-1)^{f(x)} \cdot \frac{\sum_{S \subseteq \{1,\ldots,k\}} (-1)^{\sum_{i\in S} q_i(x)}}{2^k}\right]\right| \\
&= \frac{1}{2^k}\left|\mathbb{E}_x\left[\sum_{S \subseteq \{1,\ldots,k\}} (-1)^{f(x) + \sum_{i\in S} q_i(x)}\right]\right| \\
&\leq \frac{1}{2^k} \sum_{S \subseteq \{1,\ldots,k\}} \left|\mathbb{E}\left[(-1)^{f(x) + \sum_{i\in S} q_i(x)}\right]\right| \\
&\leq \frac{2^k \varepsilon}{2^k} = \varepsilon .
\end{aligned}$$

In particular, for any variety V of size $|V| > \varepsilon 2^n$, $f(x)$ is not constant on V. $\square$

Now Theorem 5.2 and Lemma 5.4 imply the following result.[8]

Theorem 5.5. *Let $f \in B_n$ and $\varepsilon > 0$ be a constant.*

- *If* $\mathrm{Cor}(f, \omega(1)) \le 2^{-\varepsilon n}$*, then* $s_{sp}(f) = \omega(n)$*.*
- *If* $\mathrm{Cor}(f, 2^{(\log n)^{1-o(1)}}) \le 2^{-\varepsilon n}$*, then* $s_{\log}(f) = \omega(n)$*.*
- *If* $\mathrm{Cor}(f, n^{\varepsilon}) \le 2^{-\omega(n/\log\log n)}$*, then* $s_{\log}(f) = \omega(n)$*.*

5.3 Rigidity

In order to prove super-linear circuit lower bounds for log-depth circuits via Valiant's reduction, one needs to construct matrices M with rigidity $\mathbb{R}_M\left(\frac{\delta n}{\log\log n}\right) > n^{\varepsilon}$ or rigidity $\mathbb{R}_M(\varepsilon n) > 2^{(\log n)^{1-\delta}}$ for some constant $\varepsilon > 0$ and every constant $\delta > 0$. For super-linear lower bounds for series-parallel circuits, one needs to find matrices with rigidity $\mathbb{R}_M(\varepsilon n) > \delta$. Also, Razborov [Raz89] proved that rigidity $\mathbb{R}_M\left(2^{(\log\log n)^c}\right) > \frac{n}{2^{(\log\log n)^{\varepsilon}}}$ for all $c \ge 1$ gives a language that does not belong to the polynomial hierarchy for communication complexity. The best known explicit lower bound on rigidity for every r is $\mathbb{R}(r) \ge \Omega\left(\frac{n}{r}\log\frac{n}{r}\right)$ [Fri93, PV91, SSS97, Lok09].[9] Thus, for new bounds via Valiant's reduction (or Razborov's reduction for communication complexity), one needs to improve the known bounds asymptotically.

In order to get new circuit lower bounds via Theorem 1.2, we need to find a matrix $M \in \mathbb{F}^{n\times n}$ with rigidity $\mathbb{R}_M(0.75n) > 16$ (or a rectangular matrix $M \in \mathbb{F}^{m\times n}$ for $m \ge n$ which is rigid for higher rank $\mathbb{R}_M(\frac{n}{2} + \frac{m}{4}) > 16$). There are several explicit construction of matrices having rigidity $\mathbb{R}(\varepsilon n) > 16$ for some constant ε [Fri93, PV91, SSS97, Lok09]. Valiant [Val77] showed that a random matrix $M \in \mathbb{F}^{n\times n}$ has rigidity $\mathbb{R}(r) \ge \frac{(n-r)^2 - 2n - \log n}{n\log(2n^2)}$ for any $r < n - \sqrt{2n + \log n}$. In particular, $\mathbb{R}_M(n - 6\sqrt{n\log n}) > 16$ for a random matrix M. As for explicit constructions, Pudlák and Vavřín [PV91] found the exact value of rigidity (for every rank r) of the upper triangular matrix $T_n \in \mathbb{F}^{n\times n}$. In particular, they showed that $\mathbb{R}(\frac{n}{65}) > 16$. A matrix which is rigid for larger values of rank (at the price of having more outputs) was given in [PR94] and [JS13, Theorem 3.36]: A generator matrix $M \in \mathbb{F}^{m\times n}$ of a linear code with relative distance $\delta > 0$ for any $r \le n/16$ has rigidity

$$\mathbb{R}_M(r) \ge \frac{\delta n \log(n/r)}{8(r + \log(n/r))}\,.$$

We now show that using the ideas from [Fri93, SSS97], one can improve this constant, but this is still not sufficient for getting new bounds using Theorem 1.2.

[8] We remark that we do not apply these results to the depth reduction presented in this paper, but only to Valiant's depth reduction. Indeed, it would only give us a statement of the form: If $\mathrm{Cor}(f, 16) \le 2^{-n(1-\varepsilon)}$, then $s(f) \ge 3.9(1-\varepsilon)n - 4$. But as we noted in Section 1.3, every Boolean function has correlation at least $2^{-n/2}$ with some linear polynomial.

[9] There is also a semi-explicit construction due to Goldreich and Tal [GT16]. This construction can be constructed in plain-exponential time $2^{O(n)}$ and has rigidity $\mathbb{R}(r) \ge \Omega\left(\frac{n^2}{r^2\log n}\right)$ for every $r \ge \sqrt{n}$. This bound is better than the known explicit bounds for $r = o\left(\frac{n}{\log n \log\log n}\right)$. It is also known [AKTV18] how to construct a matrix with rigidity as high as $\mathbb{R}(r) \ge \Omega(n)$ for any rank $r = n^{0.5-\varepsilon}$ using subexponential time $2^{o(n)}$.

Recall that $\mathrm{H}(x) = -x \log x - (1-x)\log(1-x)$ for $0 < x < 1$, and that the generator matrix $M \in \mathbb{F}^{m \times n}$ of a code can always be transformed such that the first n rows of M form the identity matrix.

Lemma 5.6. *Let $A \in \mathbb{F}^{(m-n)\times n}$, and let $I \in \mathbb{F}^{n\times n}$ be the identity matrix. If $M = \begin{bmatrix} I \\ A \end{bmatrix}$ is a generator matrix of a linear code with relative distance δ and rate $R = \frac{n}{m}$, then $\mathbb{R}_A(r) > 16$ for*

$$r = \max_{0<\alpha<1} \left(\alpha n \cdot \mathrm{H}\left(\frac{\delta(1-\alpha)}{2\alpha(1-\alpha)R + 32\alpha} \right) \right) - o(n)\,.$$

Proof. We will show that for every 16-sparse matrix B,

$$\operatorname{rank}(A \oplus B) > \alpha n \cdot \mathrm{H}\left(\frac{\delta(1-\alpha)}{2\alpha(1-\alpha)R + 32\alpha} \right) - o(n)\,.$$

First we take the αn sparsest columns of B. By Markov's inequality, each of them has at most $\frac{16m}{(1-\alpha)n}$ non-zero entries. Let $A', B', M' \in \mathbb{F}^{m\times\alpha n}$ be the submatrices of A, B, and M corresponding to this set of αn columns. For a vector $x \in \mathbb{F}^n$, let $|x|$ be the number of non-zero elements in it.

Since M generates a code with relative distance δ, we have that for every non-zero $x \in \mathbb{F}^n$, $|Mx| \geq \delta m$. From $Mx = \begin{bmatrix} I \\ A \end{bmatrix} x = \begin{bmatrix} x \\ Ax \end{bmatrix}$, we have that $|Ax| \geq \delta m - |x|$. Since this holds for every non-zero x, including x with zeros in all coordinates *not* in A', we get that for every $x \in \mathbb{F}^{\alpha n}$, $|A'x| \geq \delta m - |x|$.

Now we only consider non-zero $x \in \mathbb{F}^{\alpha n}$ with exactly $k = \beta n$ ones where $\beta = \frac{\delta(1-\alpha)}{(1-\alpha)R+16} - o(1)$. For such an x,

$$|(A' \oplus B')x| \geq |A'x| - |B'x| \geq \delta m - |x| - |x| \cdot \frac{16m}{(1-\alpha)n} \geq \delta m - \beta n \left(1 + \frac{16m}{(1-\alpha)n}\right) > 0$$

due to the choice of β. This implies that all linear combinations of exactly $k/2$ columns from $A' \oplus B'$ are distinct. That is, the columns of $A' \oplus B'$ span at least $\binom{\alpha n}{k/2}$ points in $\mathbb{F}^m$, and

$$\begin{aligned} \operatorname{rank}(A \oplus B) \geq \operatorname{rank}(A' \oplus B') &\geq \log \binom{\alpha n}{k/2} \\ &= \alpha n \cdot \mathrm{H}(\beta/2\alpha) - o(n) \\ &= \alpha n \cdot \mathrm{H}\left(\frac{\delta(1-\alpha)}{2\alpha(1-\alpha)R + 32\alpha} \right) - o(n)\,. \end{aligned}$$

□

Let us consider Justesen's code [Jus72], [MS77, Chapter 10, §11, Theorem 12]. For $\delta = 0.077$, we have an efficient construction of a linear code with rate $R = 0.15$. In Lemma 5.6, we set $\alpha = 0.182$ and get that this matrix is rigid for rank $r > \frac{n}{64}$ beating the bound from [PV91] (at the price of having $m - n = n(1/R - 1)$ outputs).

If we take the concatenation of a Reed-Solomon code (as the outer code) and an optimal linear inner code, then for every δ we can construct in polynomial time a code with relative distance δ matching the Zyablov bound (see, e.g., the discussion in [ABN+92]):

$$R = \max_{\delta \leq \mu \leq 0.5} \left((1 - \mathrm{H}(\mu)) \left(1 - \frac{\delta}{\mu} \right) \right)\,.$$

In particular, if we take such a code with $\delta = 0.49$, then in the Zyablov bound we set $\mu = 0.493$ and get $R \approx 8 \cdot 10^{-7}$. Now we set $\alpha = 0.252$ in Lemma 5.6, and get rigidity for rank as high as $r > \frac{n}{15}$ (at the price of having too many outputs).

5.4 Open Problems

We conclude with a short summary of pseudorandom objects which would lead to new circuit lower bounds via depth reductions described in Section 4.

Open Problem 5.1. *Prove that E^{NP} contains a language f having one of the following properties:*

- *f cannot be computed by an $OR_{2^{0.8n}} \circ AND_{n \cdot 2^{15}} \circ OR_{16}$.*
- *f is a disperser for varieties of size at least $2^{0.2n}$ defined by $1.05n$ polynomials each of which depends on at most 16 variables (and, thus, has degree at most 16).*
- *f is a linear function defined by a matrix $M \in \mathbb{F}^{n \times n}$ of rigidity $\mathbb{R}_M(0.8n) > 16$ (that is, in order to decrease the rank of M to $0.8n$, one has to change more than 16 elements in some row of M).*

Open Problem 5.2. *Show that every DeMorgan formula of size s has a probabilistic polynomial over $\mathbb{F}_2$ of degree $s^{0.99}$ and error $1/3$, or give evidence this is not true. We conjecture the degree can be made $O(\sqrt{s})$.*

Acknowledgement

We thank Navid Talebanfard and Emanuele Viola for helpful discussions.

References

[ABN$^+$92] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Trans. Inf. Theory*, 38(2):509–516, 1992.

[AKTV18] Josh Alman, Mrinal Kumar, Avishay Tal, and Ben Lee Volk. Personal communication, 2018.

[And87] Alexander E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Moscow Univ. Math. Bull.*, 42(1):63–66, 1987.

[AW15] Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *FOCS 2015*, pages 136–150. IEEE, 2015.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[Bop97] Ravi B. Boppana. The average sensitivity of bounded-depth circuits. *Inf. Process. Lett.*, 63(5):257–261, 1997.

[BSG12] Eli Ben-Sasson and Ariel Gabizon. Extractors for polynomials sources over constant-size fields of small characteristic. In *RANDOM 2012*, pages 399–410, 2012.

[Cal08] Chris Calabro. A lower bound on the size of series-parallel graphs dense in long paths. In *ECCC*, volume 15, 2008.

[Cha94] Aleksandr V. Chashkin. On the complexity of Boolean matrices, graphs and their corresponding Boolean functions. *Discrete Math. and Appl.*, 4(3):229–257, 1994.

[CIP06] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *CCC 2006*, pages 252–260, 2006.

[CK15] Ruiwen Chen and Valentine Kabanets. Correlation bounds and #SAT algorithms for small linear-size circuits. In *COCOON 2015*, pages 211–222. Springer, 2015.

[CT15] Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. In *RANDOM 2015*, pages 680–709, 2015.

[CT18] Eshan Chattopadhyay and Avishay Tal. Personal communication, 2018.

[CW19] Lijie Chen and Ryan Williams. Circuit lower bounds from PCP of proximity. *Unpublished manuscript*, 2019.

[Dan96] Vlado Dančík. Complexity of Boolean functions over bases with unbounded fan-in gates. *Inf. Process. Lett.*, 57(1):31–34, 1996.

[DK11] Evgeny Demenkov and Alexander S. Kulikov. An elementary proof of a $3n - o(n)$ lower bound on the circuit complexity of affine dispersers. In *MFCS 2011*, pages 256–265, 2011.

[DKKY10] Evgeny Demenkov, Arist Kojevnikov, Alexander S. Kulikov, and Grigory Yaroslavtsev. New upper bounds on the boolean circuit complexity of symmetric functions. *Inf. Process. Lett.*, 110(7):264–267, 2010.

[DM16] Irit Dinur and Or Meir. Toward the KRW Composition Conjecture: Cubic Formula Lower Bounds via Communication Complexity. In *CCC 2016*, pages 3:1–3:51, 2016.

[Dvi12] Zeev Dvir. Extractors for varieties. *Comput. Complex.*, 21(4):515–572, 2012.

[EGS75] Paul Erdös, Ronald L. Graham, and Endre Szemerédi. On sparse graphs with dense long paths. *Comp. and Math. with Appl.*, 1:145–161, 1975.

[FGHK16] Magnus G. Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-$3n$ lower bound for the circuit complexity of an explicit function. In *FOCS 2016*, pages 89–98, 2016.

[Fri93] Joel Friedman. A note on matrix rigidity. *Combinatorica*, 13(2):235–239, 1993.

[GHKK18] Alexander Golovnev, Edward A. Hirsch, Alexander Knop, and Alexander S. Kulikov. On the limits of gate elimination. *J. Comput. Syst. Sci.*, 96:107–119, 2018.

[GK16] Alexander Golovnev and Alexander S. Kulikov. Weighted gate elimination: Boolean dispersers for quadratic varieties imply improved circuit lower bounds. In *ITCS 2016*, pages 405–411, 2016.

[GMWW14] Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: An information complexity approach to the KRW composition conjecture. In *STOC 2014*, pages 213–222, 2014.

[Gri76] Dmitrii Yu. Grigoriev. Application of separability and independence notions for proving lower bounds of circuit complexity. *Zap. Nauch. Sem. POMI*, 60:38–48, 1976.

[GT16] Oded Goldreich and Avishay Tal. Matrix rigidity of random toeplitz matrices. In *STOC 2016*, pages 91–104, 2016.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *STOC 1986*, pages 6–20, 1986.

[Hås98] Johan Håstad. The shrinkage exponent of de Morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998.

[HJP93] Johan Håstad, Stasys Jukna, and Pavel Pudlák. Top-down lower bounds for depth 3 circuits. In *FOCS 1993*, pages 124–129, 1993.

[IK17] Russell Impagliazzo and Valentine Kabanets. Fourier concentration from shrinkage. *Comput. Complex.*, 26(1):275–321, 2017.

[IN93] Russell Impagliazzo and Noam Nisan. The effect of random restrictions on formula size. *Random Struct. Algorithms*, 4(2):121–134, 1993.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[JS13] Stasys Jukna and Igor Sergeev. Complexity of linear boolean operators. *Found. Trends Theor. Comput. Sci.*, 9(1):1–123, 2013.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Trans. Inf. Theory*, 18(5):652–656, 1972.

[Khr71] Valeriy M. Khrapchenko. A method of determining lower bounds for the complexity of π-schemes. *Math. Notes of the Acad. of Sci. of the USSR*, 10(1):474–479, 1971.

[Kla94] Maria M. Klawe. Shallow grates. *Theor. Comput. Sci.*, 123(2):389–395, 1994.

[KLP12] Tali Kaufman, Shachar Lovett, and Ely Porat. Weight distribution and list-decoding size of reed–muller codes. *IEEE Trans. Inf. Theory*, 58(5):2689–2696, 2012.

[KRT13] Ilan Komargodski, Ran Raz, and Avishay Tal. Improved average-case lower bounds for demorgan formula size. In *FOCS 2013*, pages 588–597, 2013.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Comput. Complex.*, 5(3/4):191–204, 1995.

[KW90] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discrete Math.*, 3(2):255–265, 1990.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *FOCS 2016*, pages 168–177, 2016.

[Lok09] Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Found. Trends Theor. Comput. Sci.*, 4(1-2):1–155, 2009.

[Lup56] Oleg B. Lupanov. On rectifier and switching-and-rectifier schemes. *Dokl. Akad. Nauk SSSR*, 111(6):1171–1174, 1956. In Russian.

[LZ19] Fu Li and David Zuckerman. Improved extractors for recognizable and algebraic sources. In *Random*, 2019.

[MRW05] Peter Bro Miltersen, Jaikumar Radhakrishnan, and Ingo Wegener. On converting CNF to DNF. *Theor. Comput. Sci.*, 347(1-2):325–335, 2005.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes.* Elsevier, 1977.

[MW17] Or Meir and Avi Wigderson. Prediction from partial information and hindsight, with application to circuit lower bounds. In *ECCC*, volume 24, 2017.

[Nec66] Edward I. Nechiporuk. On a Boolean function. *Dokl. Akad. Nauk SSSR*, 169(4):765–766, 1966.

[PP06] Ramamohan Paturi and Pavel Pudlák. Circuit lower bounds and linear codes. *J. Math. Sci.*, 134(5):2425–2434, 2006.

[PPSZ05] Ramamohan Paturi, Pavel Pudlák, Michael E Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, 2005.

[PPZ97] Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. In *FOCS 1997*, pages 566–574, 1997.

[PR94] Pavel Pudlák and Vojtech Rödl. Some combinatorial-algebraic problems from complexity theory. *Discrete Math.*, 136(1-3):253–279, 1994.

[PSZ97] Ramamohan Paturi, Michael E. Saks, and Francis Zane. Exponential lower bounds for depth 3 Boolean circuits. In *STOC 1997*, pages 86–91, 1997.

[PV91] Pavel Pudlák and Zdeněk Vavřín. Computation of rigidity of order $\frac{n^2}{r}$ for one simple matrix. *Comment. Math. Univ. Carolinae*, 32(2):213–218, 1991.

[PZ93] Mike Paterson and Uri Zwick. Shrinkage of de Morgan formulae under restriction. *Random Struct. Algorithms*, 4(2):135–150, 1993.

[Raz87] Alexander A. Razborov. Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Mat. Zametki*, 41(4):598–607, 1987.

[Raz89] Alexander A. Razborov. On rigid matrices. *Manuscript*, 1989. In Russian.

[Rei11] Ben W. Reichardt. Reflections for quantum query algorithms. In *SODA 2011*, pages 560–569. SIAM, 2011.

[Rem16] Zachary Remscrim. The Hilbert function, algebraic extractors, and recursive fourier sampling. In *FOCS 2016*, pages 197–208, 2016.

[San10] Rahul Santhanam. Fighting perebor: New and improved algorithms for formula and QBF satisfiability. In *FOCS 2010*, pages 183–192, 2010.

[Sch82] Georg Schnitger. A family of graphs with expensive depth-reduction. *Theor. Comput. Sci.*, 18(1):89–93, 1982.

[Sch83] Georg Schnitger. On depth-reduction and grates. In *FOCS 1983*, pages 323–328, 1983.

[Sch05] Rainer Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *J. Algorithms*, 54(1):40–44, 2005.

[Ser18] Igor S. Sergeev. On complexity of circuits and formulas of bounded depth over unbounded fan-in bases. *Disc. Math. Appl.*, 30(2):120–137, 2018. In Russian.

[Sha49] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell Syst. Tech. J.*, 28:59–98, 1949.

[She12] Alexander A Sherstov. Making polynomials robust to noise. In *STOC 2012*, pages 747–758. ACM, 2012.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *STOC 1987*, pages 77–82, 1987.

[SS12] Rahul Santhanam and Srikanth Srinivasan. On the limits of sparsification. In *ICALP 2012*, pages 774–785, 2012.

[SSS97] Mohammad Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. A remark on matrix rigidity. *Inf. Process. Lett.*, 64(6):283–285, 1997.

[ST13] Kazuhisa Seto and Suguru Tamaki. A satisfiability algorithm and average-case hardness for formulas over the full binary basis. *Comput. Complex.*, 22(2):245–274, 2013.

[Sub61] Bella A. Subbotovskaya. Realizations of linear functions by formulas using $+, \cdot, -$. *Dokl. Akad. Nauk SSSR*, 136(3):553–555, 1961.

[Tal14] Avishay Tal. Shrinkage of De Morgan formulae by spectral techniques. In *FOCS 2014*, pages 551–560. IEEE, 2014.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *MFCS 1977*, pages 162–176, 1977.

[Vio07] Emanuele Viola. Pseudorandom bits for constant-depth circuits with few arbitrary symmetric gates. *SIAM J. Comput.*, 36(5):1387–1403, 2007.

[Vio09] Emanuele Viola. On the power of small-depth computation. *Found. Trends Theor. Comput. Sci.*, 5(1):1–72, 2009.

[VW08] Emanuele Viola and Avi Wigderson. Norms, XOR lemmas, and lower bounds for polynomials and protocols. *Theory Comput.*, 4(1):137–168, 2008.

[Wil18] Richard Ryan Williams. Limits on representing Boolean functions by linear combinations of simple functions: Thresholds, ReLUs, and low-degree polynomials. In *CCC 2018*, pages 6:1–6:24, 2018.

[Wol06] Guy Wolfovitz. The complexity of depth-3 circuits computing symmetric Boolean functions. *Inf. Process. Lett.*, 100(2):41–46, 2006.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il