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

Circuit Depth Reductions

Alexander Golovnev* Alexander S. Kulikov†

Abstract

The best known circuit lower bounds against unrestricted circuits 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 suggest a first non-gate-elimination approach for obtaining circuit lower bounds. Namely, we prove that every (unbounded-depth) circuit of size s can be expressed as an OR of $2^{s/3.9}$ 16-CNFs. While this structural result does not immediately lead to new lower bounds, it suggests a new avenue of attack on them.

Our result complements the classical depth reduction result of Valiant which asserts that *logarithmic*-depth circuits of linear size can be computed by an OR of $2^{\varepsilon n}$ n^{δ}-CNFs. It is known that no such graph-theoretic reduction can work for circuits of super-logarithmic depth. We overcome this limitation (for small circuits) by taking into account both the graph-theoretic and functional properties of circuits.

We show that *qualitative* improvements of the following pseudorandom constructions imply *qualitative* (from $3n$ to $\omega(n)$) improvement of size 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, now even modest *quantitative* improvements of the known constructions give elementary proofs of *quantitatively* stronger circuit lower bounds ($3.9n$ instead of $3n$).

1 Introduction

1.1 Circuits

Boolean circuits is a natural model for computing Boolean functions. A circuit corresponds to a simple straight line program where every instruction performs a binary Boolean operation on two operands each of which is either an input variable or a result of some previous instruction. The structure of this program is extremely simple: no loops, no conditional statements. Still, we have no example of a function from P (or even NP, or even E^{NP}) that requires at least $3.1n$ binary instructions to compute (let alone superlinear or superpolynomial size). This is in sharp contrast with the fact that finding such a function non-constructively is easy. For this, one compares the number 2^{2^n} of different functions of n variables with the number of programs of a fixed size. One then concludes, and this was done by Shannon [Sha49] some seventy years ago, that a random function on n variables has circuit size $\Omega(2^n/n)$ with probability $1-o(1)$. This bound was later proven to be tight by Lupanov [Lup59]: any function can be computed by a circuit of size about $2^n/n$.

*Harvard University, email: alexgolovnev@gmail.com

†Steklov Institute of Mathematics at St. Petersburg, email: alexanderskulikov@gmail.com

ISSN 1433-8092

The strongest known circuit size lower bound $(3 + \frac{1}{86})n - o(n)$ is proven for affine dispersers for sublinear dimension [FGHK16]. This proof, as well as all previous proofs, is based on the gate elimination technique. Its 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 for proving lower bounds for unrestricted circuits, the stronger is the bound the more tedious is its case analysis: when eliminating, say, three or four gates one has to go through all possible cases when two of these gates coincide. It is currently difficult to imagine a proof of $5n$ lower bound using these ideas. This was recently made formal in [GHKK18] where it was shown that a certain formalization of the gate elimination technique is unable to get a stronger than $5n$ lower bounds. Thus, we need to find new approaches for proving lower bounds against circuits of unbounded depth.

1.2 Linear Circuits

Superlinear lower bounds are not known even for linear circuits, i.e., circuits consisting of $\oplus$ gates only. Every linear function with one output can be computed by a circuit of size $n - 1$. Thus, for linear circuits, it makes sense to consider multi-output functions of the form $f(x) = Ax$ where $A \in \{0,1\}^{m \times n}$. Again, 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 explicit matrices the strongest known lower bound is $3n - o(n)$ due to Chashkin [Cha94]. Interestingly, the proof of Chashkin [Cha94] 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 any circuit for it has at least $n - o(n)$ gates of out-degree at least 2.[1] Then he "pads" it to an $n \times n$ matrix and shows that $n - o(n)$ additional gates are needed. Similarly, the best known lower bound on the complexity of linear circuit with $\log n \leq m < o(n^2)$ outputs is $2n + m - o(n)$.

1.3 Log-Depth Circuits

Nothing stronger than $(3 + \frac{1}{86})n - o(n)$ is known even if we restrict the depth of a circuit to be $O(\log n)$. It is straightforward to show that any function that depends essentially 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 [Nec66]. Still, proving a superlinear size lower bound for circuits of depth $O(\log n)$ is a major open problem [Val77].

1.4 Constant-Depth Circuits

Another natural and simple model of computation is bounded depth circuits that correspond to highly parallelizable computations. 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). The usual assumption is that the inputs of the circuit are variables and their negations, and the fan-in of the gates is unbounded. Such circuits are much more structured and therefore are easier to analyze and to prove lower bounds, in particular. For example, it is easy to show that the minimal number of clauses in a CNF computing the parity function of n bits is equal to 2^{n-1}, which gives an optimal lower bound on the size of depth-2 circuits. However, already for depth 3 there is again a large

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

gap between known lower and upper bounds: whereas it was shown by Lupanon [Lup61] that the minimum depth-3 circuit size of a random function on n variables is $\Theta(2^n/n)$, 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 however for depth-3 circuits where the fan-in of the gates that are close to the inputs is bounded by k. Namely, for any $k = O(\sqrt{n})$, a $2^{n/k}$ lower bound is proven by Paturi, Saks, and Zane [PPZ97] for the parity function, and a 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].

Calabro, Impagliazzo, and Paturi [CIP06] construct a family of $2^{O(n^2)}$ functions most of which require depth-3 circuits 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 $f(n) = \omega(n \log n)$.

1.5 Valiant's Depth Reduction

Remarkably, the classical result of Valiant from the 70's relates the three computational models described above. Using a depth reduction for DAGs [EGS75], Valiant [Val77] shows that in any circuit of size cn and depth d, for every integer k, one can remove $\frac{2ckn}{\log d}$ wires, such that the resulting circuit has depth $d/2^k$. Valiant concludes that if the depth of the resulting circuit is non-trivial $d/2^k < \log n$, then a lower bound on depth-3 circuits implies a lower bound against the original circuit model. This way, Valiant shows that any circuit of size $O(n)$ and depth $O(\log n)$ can be converted into an OR $\circ$ AND $\circ$ OR circuit with the fan-in of the output gate at most $2^{O(n/\log\log n)}$ and the fan-in of OR-gates fed by the inputs at most $n^{O(1)}$. Hence, by exhibiting an explicit function that has no depth-3 circuit with these restrictions, one immediately gets that this function cannot be computed by circuits of linear size and logarithmic depth. Unfortunately, the known lower bounds on depth-3 circuits (see Subsection 1.4) are still too far from the ones 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 logarithmic 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 linear circuits of logarithmic depth.

1.6 Motivating Example

For Valiant's depth reduction, one can have $d/2^k < \log n$ (and non-trivial number of removed edges $\frac{2ckn}{\log d}$) only for circuits of depth $d = O(\log n)$. Thus, Valiant's depth reduction technique does not apply to circuits of larger depth. Moreover, Schnitger and Klawe [Sch82, Sch83, Kla94] construct an explicit family of DAGs showing that the parameters achieved by Valiant are essentially optimal. This counter-example shows that the 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. We

give more details in the next subsection, and here we provide a simple example showing that such a reduction is possible in principle.

By a formula we mean a circuit where each gate has out-degree exactly 1 (for the ease of exposition, here we consider formulas where gates can compute arbitrary binary Boolean operations). Here we 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 an almost quadratic lower bound [Nec66] on formula size, we could hope to find a function which cannot be computed by an OR of fewer than 2^n linear-size formulas.

Lemma 1.1. *A circuit $\mathcal{C}$ of size s can be computed by an OR of $2^{\lceil s/3 \rceil}$ formulas each of size at most s.*

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

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

Using the induction hypothesis for $\mathcal{C}_0$ and $\mathcal{C}_1$, we 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 $2^{-n+o(n)}$. Actually, there is an inherent limitation for this approach. By Parseval's inequality, 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). Every parity function can be computed by a circuit of size $\leq n$, thus, Lemma 1.1 would only be able to prove circuit lower bounds of $1.5n$.

Therefore, 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. In the following subsection, we describe a reduction that 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.

1.7 Our Contribution

The main contribution of this paper is counterparts of Valiant's reduction, but for unrestricted circuits. They are summarized[2] in Table 1. We highlight several important properties of the presented depth reduction techniques below.

[2]In this table we only present strongest implications from the strongest premises. In fact, our reductions would give new circuit lower bounds even from weaker objects (see Section 4 for formal statements of the results). For

	improving known lower bound	to lower bound	implies new 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.	$s_{\log}(f) = \omega(n)$
*	$(16, \infty, 2^{(1-\varepsilon)n})$-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	$\mathrm{Cor}(f, n^\varepsilon) \leq 2^{-n^{1/2-\varepsilon}}$ [Rem16]	$\mathrm{Cor}(f, n^\varepsilon) \leq 2^{-\omega\left(\frac{n}{\log\log n}\right)}$	$s_{\log}(f) = \omega(n)$
*	$\mathrm{Cor}(f, 16) \leq 2^{-\varepsilon n}$ [VW08]	$\mathrm{Cor}(f, 16) \leq 2^{-n+o(n)}$	$s(f) \geq 3.9n$
V	$\mathcal{R}_M\left(\omega\left(\frac{n}{\log\log n}\right)\right) > \log n$ [Fri93]	$\mathcal{R}_M\left(\omega\left(\frac{n}{\log\log n}\right)\right) > n^\varepsilon$	$s_{\oplus,\log}(M) = \omega(n)$
*	$\mathcal{R}_M(\frac{n}{65}) > 16$ [PV91]	$\mathcal{R}_M(n - o(n)) > 16$	$s_\oplus(M) \geq 4n$

Table 1: Comparing the depth reduction results of this paper (labeled with *) with the results of Valiant [Val77] (labeled with V). We use the following notation (all formal definitions are given in Sections 2 and 4): $s(f)$ 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; $\mathrm{Cor}(f, d)$ is the correlation of f with polynomials of degree d; $\mathcal{R}_M(r)$ is the row-rigidity of M for the rank r, i.e., the smallest row-sparsity of a matrix A such that $\mathrm{rank}(M \oplus A) \leq r$.

Easier to achieve. In order to get a new circuit lower bound through Valiant's depth reduction, one needs to achieve a *qualitative* improvement of known lower bounds for depth-3 circuits or matrix rigidity (see the table). In a sense, Valiant's result states that an asymptotic improvement in one direction implies an asymptotic improvement in the other one. In contrast, our depth reduction shows that a *quantitative* improvements of known lower bounds imply modest improvements of circuit lower bounds. Thus, improvements required by our reductions, are probably easier to achieve.

Unrestricted depth. As already mentioned, Valiant's reduction works for circuits of logarithmic depth. Our reduction works for circuits of any depth, though they are only meaningful when the circuit has modest linear size.

Not just graph-based. Graph-theoretic approaches (e.g., Valiant's technique) are inherently limited to circuits of logarithmic depth. We are able to work with unrestricted depth since our

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 $3.1n$.

approach is based not just on the underlying graph of a circuit, but also on the actual computations happening inside the circuit.

No case analysis. Unlike gate elimination proofs of circuit lower bounds, our depth reductions contain almost no case analysis.

No known limitations. The strongest known lower bounds for circuits, as well as most of the previous lower bounds, are proven by the gate elimination technique. This technique is known to be too weak to prove a $5n$ lower bound [GHKK18]. The corresponding limitation does not apply to the depth reduction presented in this paper. We remark that this limitation also does not apply to the approach based on efficient SAT-algortihms [Wil13, BSV14, JMV15].

New structural questions. On the way of proving new lower bounds, we also study structural results on converting small circuits into ORs of k-CNFs that have curious connections to various properties of k-CNFs (such as guaranteed by Satisfiability Coding Lemma [PPZ97, PPSZ05] and Sparsification Lemma [IPZ01, CIP06]).

Hierarchy of pseudorandom objects. We also show that improvements on the known constructions of pseudorandom objects (dispersers for varieties, functions with small correlation with low degree polynomials, and rigid matrices) immediately imply stronger circuit lower bounds via presented depth reductions (and significant improvements on these constructions imply strong bounds via Valiant's depth reduction).

2 Definitions

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 $s_{\oplus}$.

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 $\mathrm{OR} \circ \mathrm{AND} \circ \mathrm{OR}$ 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 $\mathrm{OR}_p \circ \mathrm{AND}_q \circ \mathrm{OR}_r$ 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 $\mathrm{OR} \circ \mathrm{AND} \circ \mathrm{OR}_k$ 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 $\mathrm{OR} \circ \mathrm{AND} \circ \mathrm{OR}_k$ circuit computing f.

2.4 Rigidity

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

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

3 Depth Reductions

In this section, we present new depth reductions for circuits with unrestricted depth. First, we present the classical depth reduction results by Valiant [Val77].

Theorem 3.1 ([Val77, Cal08, Vio09])**.** *For every $c, \varepsilon > 0$ there exists $\delta > 0$ such that any circuit $\mathcal{C}$ of size cn and depth $c \log n$ can be computed as*

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

A series-parallel circuit of size cn and unbounded depth can be computed as an $OR_{2^{\varepsilon n}} \circ AND \circ OR_\delta$ circuit.

This result applied to linear circuits gives the following theorem.

Theorem 3.2 ([Val77, Cal08, Vio09])**.** *Let $M \in \{0,1\}^{m \times n}$ be a matrix. For every $c, \varepsilon > 0$ there exists $\delta > 0$ such that if a linear circuit $\mathcal{C}$ of size cn and depth $c \log n$ computes Mx for every $x \in \{0,1\}^n$, then*

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

If $\mathcal{C}$ is a series-parallel linear circuit of size cn and unbounded depth, then $\mathcal{R}_M(\varepsilon n) \leq \delta$.

3.1 Linear Circuits

In this subsection, we deal with *linear* circuits, i.e., circuits consisting of $\oplus$ gates only. For technical reasons, we assume that there are $n+1$ input gates of a linear circuit: $x_1, \ldots, x_n$ as well as 0. For a matrix $M \in \{0,1\}^{m \times n}$, we say that a linear circuit computes the linear transformation Mx (or just the matrix M itself), if some m gates of the circuit are labeled as outputs and the i-th output computes the linear sum of the subset of n input variables specified by the i-th row of M (in particular, if a row of M has at most one 1, then the corresponding output label is placed on the corresponding input gate). When the m output gates of $\mathcal{C}$ are specified, for $x \in \{0,1\}^n$, we treat $\mathcal{C}(x)$ as the vector of output values. Then, $\mathcal{C}$ computes M if $\mathcal{C}(x) = Mx$ for all $x \in \{0,1\}^n$. We say that a linear circuit $\mathcal{C}$ computing M is *optimal* if no other circuit of smaller size computes M.

The main result of this subsection asserts that matrices computable by small circuits are not too rigid. The contrapositive of this statement is: to get an improved lower bound on the size of linear circuits, it suffices to construct a matrix with good rigidity parameters.

Theorem 3.3. *Let $M \in \{0,1\}^{m \times n}$ and let $\mathcal{C}$ be a linear circuit of size s computing M. Then*

$$\mathcal{R}_M(\lfloor s/4 \rfloor) \leq 16\,.$$

Proof. 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 induction on s. For the induction step, we assume further that $\mathcal{C}$ is optimal (if it is not, the statement holds just by the induction hypothesis).

We now "normalize" the circuit $\mathcal{C}$. Namely, our goal is to show that the matrix M can be decomposed into the sum $A \oplus B$ where the matrix A is 16-sparse and the matrix B has rank at most $\lfloor s/4 \rfloor$. Now, if $\mathcal{C}$ has an output gate H of depth at most 4 (recall that the depth of a gate is the maximum number of wires on a path from the gate to an input gate), then H computes a linear function that depends on at most 16 input variables. This, in turn, means that the corresponding row of M has at most 16 ones. Consider now the matrix M_H resulting from M by removing the corresponding row. It is not difficult to see that $\mathcal{R}_{M_H}(\lfloor s/4 \rfloor) \leq 16$ implies $\mathcal{R}_M(\lfloor s/4 \rfloor) \leq 16$. Indeed, assume that $M_H = A_H \oplus B_H$ where A_H is 16-sparse and $\mathrm{rank}(B_H) \leq \lfloor s/4 \rfloor$. To get the same decomposition for M, we add to M_H and A_H the removed row and we add the 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 that $\mathcal{C}$ has no output gates of depth at most 4.

Claim 3.4. *Let $\mathcal{C}$ be an optimal linear circuit computing $M \in \{0,1\}^{m \times n}$ such that $s(\mathcal{C}) \geq 16$, and no output gate of $\mathcal{C}$ has depth smaller than 5. Then $\mathcal{C}$ contains a gate G such that there exists a linear circuit $\mathcal{C}'$ computing $M' \in \{0,1\}^{m \times n}$ (i.e., of exactly the same size as M) such that*

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

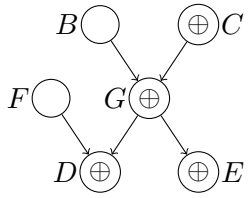
Consider the circuit $\mathcal{C}'$ and the matrix M' provided by Claim 3.4. Let $g \in \{0,1\}^{1 \times n}$ be a characteristic vector of the linear function computed at the gate G by the circuit $\mathcal{C}$: $G(x) = gx$. We know that $gx = 0$ implies $(M \oplus M')x = 0$. Hence $(M \oplus M')$ is either zero or defines exactly 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 $\mathrm{rank}(B') \leq \lfloor \frac{s-4}{4} \rfloor = \lfloor \frac{s}{4} \rfloor - 1$. 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.

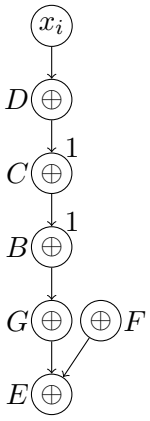
□

Proof of Claim 3.4.

Case 1: there exists a gate in $\mathcal{C}$ that has depth at least 2 and at most 4 and has out-degree at least 2. Call it G, call its predecessors B and C, and call two of its successors D and E, see Figure 3.1 (in this and following figures we write the out-degrees of some of the gates near them). The circuit $\mathcal{C}'$ is obtained from $\mathcal{C}$ by simplifying $\mathcal{C}$ using $G = 0$. Indeed, the gate G is not needed in $\mathcal{C}'$. Also, $B(x) = C(x)$ 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: under $G = 0$, the gate G is removed, B is replaced by C, D and E are replaced by their other predecessors.



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

Figure 1: Cases in the proof of Claim 3.4.

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 follow 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. Note that the gates B and C have out-degree 1. This essentially means that in $\mathcal{C}$ they are used for computing the gate G only. This, in turn, means that under

$G = 0$ one removes G, B, and C (none of them is an output). Also, the gate E is replaced by the other input F of E ($F \neq B, C, G$ since $\mathcal{C}$ is optimal).

□

Remark 3.5. *Using similar ideas, one can show that any linear circuit 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 optimal circuits $\mathcal{C}_0$ and $\mathcal{C}_1$ resulting from $\mathcal{C}$ by simplifying it under $G = 0$ and $G = 1$, respectively. As shown in the proof, both these circuits have size at most $s - 4$. One then proceeds by induction. We illustrate this approach in full detail in the next subsection.*

Remark 3.6. *The proof of Theorem 3.3 gives a decomposition $M = A \oplus B = A \oplus C \cdot D$, where $A \in \{0,1\}^{m \times n}$ is 16-sparse, $C \in \{0,1\}^{m \times s/4}$ is composed of vectors t, and $D \in \{0,1\}^{s/4 \times n}$ is composed of vectors g. Since the chosen gate G always has depth at most four, the vector g is 16-sparse. Thus, we have a decomposition $M = A \oplus C \cdot D$, where A and D are both 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.*

3.2 General Circuits

In this section, we study the following natural question: given a circuit[3], 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 3.7. *A 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. This implies that an $\mathrm{OR}_p \circ \mathrm{AND}_q \circ C(r)$ circuit can be easily converted into a $\mathrm{OR}_p \circ \mathrm{AND}_{q2^r} \circ \mathrm{OR}_{r+1}$ circuit. This way, we get the following corollary from Theorem 3.7.

Corollary 3.8. *A circuit of size s can be computed as:*

1. *an $OR_{2^{\lceil \frac{s}{2} \rceil}} \circ AND_s \circ OR_2$ circuit;*
2. *an $OR_{2^{\lceil \frac{s}{3.9} \rceil}} \circ AND_{2^{14} \cdot s} \circ OR_{16}$ circuit.*

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

Hence, $\alpha(2) \leq \frac{1}{2}$ *and* $\alpha(16) \leq \frac{1}{3.9}$.

Proof of Theorem 3.7. 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 that both $\mathcal{C}_i$'s have the smallest possible 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 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 $\mathrm{OR}_{p_i} \circ \mathrm{AND}_{q+1} \circ C(\max\{r_i, t\})$ 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).

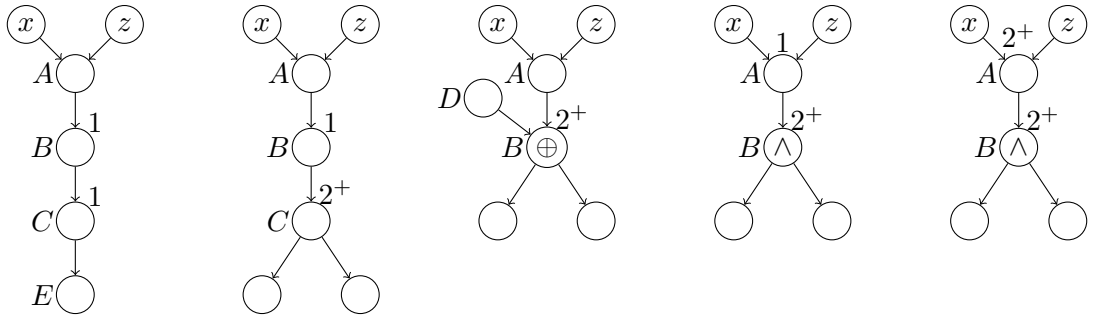
1. The base case is $s = 1$. Then $\mathcal{C}$ consists of a single gate and can be expressed as an $\mathrm{OR}_1 \circ \mathrm{AND}_1 \circ C(1)$ circuit. For the induction step, assume that $s \geq 2$ and take a gate A that depends on two variables. Let $G = A$, hence $t = 1$. The gate A must have 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. Denote the variables that feed A by x and z and 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 $C(G) \leq 15$ 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: 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 3.7.

Case 1.1: $\text{out}(C) = 1$. 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: $\text{out}(C) \geq 2$. Let $G = C$. In $\mathcal{C}_i$'s, one removes B, C, and the successors of C.

Case 2: $\text{out}(B) \geq 2$. 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 2.1: $B(A, D) = A \oplus D \oplus c$ where $c \in \{0, 1\}$. Let $G = C$. 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: $\text{out}(A) = 1$. Let $G = C$. In $\mathcal{C}_i$'s, one removes B, its successors, and A.

Case 2.2.2: $\text{out}(A) \geq 2$. 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 $2^{-\frac{3}{3.9}} + 2^{-\frac{5}{3.9}} < 1$.

□

Remark 3.9. *It is not difficult to see that the output OR gate can be replaced by a SUM gate over the integers. In other words, for any $x \in \{0,1\}^n$, at most one of the subcircuits feeding the OR gate may evaluate to 1. This holds because we always consider two mutually exclusive cases: $G = 0$ or $G = 1$.*

3.3 Properties of $\alpha(k)$

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

Lemma 3.10. *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], any k-CNF has at most $2^{n(1-1/k)}$ isolated satisfying assignments. This implies that f 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 $\alpha(2) = 2$, they must have circuit complexity at least $2n - o(n)$.

One can use techniques from Theorem 3.7 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 3.11.

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

Proof.

1. The function IP is known to be a disperser for projections for dimension $d = \frac{n}{2}+1$ (see, e.g., [CS16, Theorem A.1]). This means that it does not degenerate to a constant after any $n-d$ substitutions (called projections) of the form $x_i \leftarrow x_j \oplus c$ where $c \in \{0,1\}$. For such dispersers, an $2^{n-d-o(n)} = 2^{\frac{n}{2}-o(n)}$ lower bound on s_3^2 is proven by [PSZ97]. The upper bound follows from the fact that $\mathrm{IP}(x_1,\ldots,x_n) = 1$ iff there is an odd number of 1's among

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

Hence,

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

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

2. The lower bound is a direct consequence of the lower bound $s_3^3(\oplus_n) \geq 2^{\frac{n}{3}}$ (by substituting every second input of IP by 1, one gets the function $\oplus_{\frac{n}{2}}$).

   For the 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 $p_1,\ldots,p_{\frac{n}{4}}$ such that an odd number of them is equal to 1:

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

   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:

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

   The fan-in of the resulting OR-gate is

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

□

Open Problem 3.1. *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.

Open Problem 3.2. *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 where $\delta(c) > 0$.

For example, we can consider one of the classes 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] and that $s_3^2(f) \leq \mathrm{poly}(n) \cdot 1.5^n$ [PSZ97]. For $0 < x < 1$, let $\mathrm{H}(x) = -x \log x - (1-x)\log(1-x)$ be the binary entropy function. Generalizing the result of [PSZ97], one gets an upper bound $s_3^k(f) \leq 2^{\beta(k)n}$, where

$$\beta(k) = \max_{0<a<0.5} H(a) - \frac{a}{\lfloor \frac{k}{2} \rfloor} \cdot \log \binom{k}{\lfloor \frac{k}{2} \rfloor}.$$

For every fixed k, it is trivial to find this maximum and $\beta(k)$. In particular, we have $s_3^3(f) \leq \mathrm{poly}(n) \cdot \left(\frac{4}{3}\right)^n$, and $s_3^\infty(f) \leq \mathrm{poly}(n) \cdot \left(\frac{5}{4}\right)^n$ for any symmetric function f.[4]

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 3.12 ([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 Cn 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 3.12 implies that $\alpha(k, c) \geq \alpha(\infty, c) \geq \alpha(k, c) - O\big(\frac{\log(ck)}{k}\big)$.

4 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);

[4] Here by an upper bound on s_3^∞ we denote the expression $(\lim_{k\to\infty} 2^{\beta_k})^n$.

- 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.

4.1 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 bounds [GK16]. Now we show that dispersers for varieties of degree 16 will give new circuit lower bounds via a new simple method.

Definition 4.1. *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 4.2. *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 4.1 (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 4.2. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a function with $|f^{-1}(1)| \ge |f^{-1}(0)|$ and $\varepsilon > 0$ be a constant.*[5]

- *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.

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

- From Theorem 3.7, 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 $f_1, \ldots, f_t \colon \{0,1\}^n \to \{0,1\}$ 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,

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

  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 $f_i^{-1}(1) \leq 2^{\varepsilon n}$. Now, (4) implies that $s/3.9 \geq n - \varepsilon n - 1$.

- 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 3.1, 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 4.1, 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 $f_1, \ldots, f_t \colon \{0,1\}^n \to \{0,1\}$ 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 4.2, 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, LZ18]. 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) [LZ18]. 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 3.7.

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

Lemma 4.3. *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 $1 - o(1)$. □

4.2 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 4.3. *For two functions $f, g \colon \{0,1\}^n \to \{0,1\}$, 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 $\{0,1\}^n$.

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 4.4 (Implicit in [Dvi12, CT18, LZ18])**.** *If $\mathrm{Cor}(f, d) \leq \varepsilon$, then f is $(d, \infty, \varepsilon \cdot 2^n)$-disperser.*

Proof. Consider a variety $S = \{x \in \{0,1\}^n \colon q_1(x) = \cdots = q_k(x) = 0\}$, where each $q_i \colon \{0,1\}^n \to \{0,1\}$ 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 S, and from the Fourier expansion we have

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

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 S of size $|S| > \varepsilon 2^n$, $f(x)$ is not constant. □

Now Theorem 4.2 and Lemma 4.4 imply the following result.

Theorem 4.5. *Let $f \in B_n$ and $\varepsilon > 0$ be a constant.*

- *If* $\mathrm{Cor}(f, 16) \leq 2^{-n(1-\varepsilon)}$, *then* $s(f) \geq 3.9(1-\varepsilon)n - 4$.
- *If* $\mathrm{Cor}(f, \omega(1)) \leq 2^{-\varepsilon n}$, *then* $s_{sp}(f) = \omega(n)$.
- *If* $\mathrm{Cor}(f, 2^{(\log n)^{1-o(1)}}) \leq 2^{-\varepsilon n}$, *then* $s_{\log}(f) = \omega(n)$.
- *If* $\mathrm{Cor}(f, n^{\varepsilon}) \leq 2^{-\omega(n/\log\log n)}$, *then* $s_{\log}(f) = \omega(n)$.

4.3 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 $\mathcal{R}_M\left(\frac{\delta n}{\log\log n}\right) > n^{\varepsilon}$ or rigidity $\mathcal{R}_M(\varepsilon n) > 2^{(\log n)^{1-\delta}}$ 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 $\mathcal{R}_M(\varepsilon n) > \delta$. Also, Razborov [Raz89] proved that rigidity $\mathcal{R}_M\left(2^{(\log\log n)^{\delta}}\right) > \frac{n}{2^{(\log\log n)^{\varepsilon}}}$ 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 $\mathcal{R}(r) \geq \Omega\left(\frac{n}{r}\log\frac{n}{r}\right)$ [Fri93, PV91, SSS97, Lok09].[6] 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 3.3, we need to find a matrix $M \in \{0,1\}^{n\times n}$ with rigidity $\mathcal{R}_M(0.75n) > 16$ (or a rectangular matrix $M \in \{0,1\}^{m\times n}$ for $m \geq n$ which is rigid for higher rank $\mathcal{R}_M(\frac{n}{2}+\frac{m}{4}) > 16$). There are several explicit construction of matrices having rigidity $\mathcal{R}(\varepsilon n) > 16$ for some constant ε [Fri93, PV91, SSS97, Lok09]. Valiant [Val77] showed that a random matrix $M \in \{0,1\}^{n\times n}$ has rigidity $\mathcal{R}(r) \geq \frac{(n-r)^2-2n-\log n}{\log(2n^2)}$ for any $r < n-\sqrt{2n+\log n}$. In particular, $\mathcal{R}_M(n-2\sqrt{n}) \gg 16$ 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 $T_n \in \{0,1\}^{n\times n}$. In particular, they showed that $\mathcal{R}(\frac{n}{65}) > 16$. 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 $M \in \{0,1\}^{m\times n}$ of a linear code with relative distance $\delta > 0$ for any $r \leq n/16$ has rigidity

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

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 3.3.

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

Lemma 4.6. *Let $A \in \{0,1\}^{(m-n)\times n}$, and let $I \in \{0,1\}^{n\times n}$ 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 $\mathcal{R}_A(r) > 16$ 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,

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

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 $A', B', M' \in \{0,1\}^{m\times\alpha n}$ be the submatrices of A, B, and M corresponding to this set of αn columns. For a vector $x \in \{0,1\}^n$, 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 $x \in \{0,1\}^n$, $|Mx| \geq \delta m$. 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 $x \in \{0,1\}^{\alpha n}$, $|A'x| \geq \delta m - |x|$.

[6] There is also a semi-explicit construction due to Goldreich and Tal [GT16]. This construction uses $O(n)$ random bits and has rigidity $\mathcal{R}(r) \geq \Omega\left(\frac{n^2}{r^2\log n}\right)$ for every $r \geq \sqrt{n}$. This bound is better than the known explicit bounds for $r = o\left(\frac{n}{\log n \log\log n}\right)$.

Now we only consider non-zero $x \in \{0,1\}^{\alpha n}$ 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 $\{0,1\}^m$, 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 4.6, 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 4.6, and get rigidity for rank as high as $r > \frac{n}{15}$ (at the price of having too many outputs).

4.4 Open Problems

In this section we give a short summary of pseudorandom objects which would lead to new circuit lower bounds via depth reductions described in Section 3.

Open Problem 4.1. *Prove that E^{NP} contains a language f having one of the following properties:*

- *f cannot be computed by an $OR_{2^{0.2n}} \circ AND_{n \cdot 2^{15}} \circ OR_{16}$.*
- *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 has correlation at most $2^{-0.2n}$ with polynomials of degree 16.*
- *f is a linear function defined by a matrix $M \in \{0,1\}^{n \times n}$ of rigidity $\mathcal{R}_M(0.8n) > 16$ (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).*

Acknowledgements

We would like to thank Ryan Williams for fruitful discussions on this topic.

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.

[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.

[BSV14] Eli Ben-Sasson and Emanuele Viola. Short PCPs with projection queries. In *ICALP 2014*, pages 163–173, 2014.

[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.

[CS16] Gil Cohen and Igor Shinkar. The complexity of DNF of parities. In *ITCS 2016*, pages 47–58, 2016.

[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.

[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.

[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.

[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.

[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.

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

[JMV15] Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. In *ICALP 2015*, pages 749–760, 2015.

[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.

[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.

[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. In *Dokl. Akad. Nauk SSSR*, volume 111, pages 1171–1174, 1956. In Russian.

[Lup59] Oleg B. Lupanov. A method of circuit synthesis. *Izv. VUZov, Radiofizika*, 1:120–140, 1959. In Russian.

[Lup61] Oleg B. Lupanov. On realization of functions of propositional calculus by formulas of bounded depth over the basis $\{\&, \vee, \neg\}$. In *Dokl. Akad. Nauk SSSR*, volume 136, pages 1041–1042, 1961. In Russian.

[LZ18] Fu Li and David Zuckerman. Improved extractors for recognizable and algebraic sources. In *ECCC*, volume 25, 2018.

[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.

[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.

[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.

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

[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.

[Tal14] Avishay Tal. Shrinkage of de morgan formulae by spectral techniques. In *FOCS 2014*, pages 551–560, 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. *Foundations and Trends in Theoretical Computer Science*, 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.

[Wil13] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM J. Comput.*, 42(3):1218–1244, 2013.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il