Circuit Lower Bounds for MCSP from Local Pseudorandom Generators

Mahdi Cheraghchi
Department of Computing, Imperial College London, London, UK
http://mahdi.ch
m.cheraghchi@imperial.ac.uk

Valentine Kabanets
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
https://www.cs.sfu.ca/~kabanets/
kabanets@cs.sfu.ca

Zhenjian Lu
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
zhenjian_lu@sfu.ca

Dimitrios Myrisiotis
Department of Computing, Imperial College London, London, UK
d.myrisiotis17@imperial.ac.uk

Abstract

The Minimum Circuit Size Problem (MCSP) asks if a given truth table of a Boolean function f can be computed by a Boolean circuit of size at most θ, for a given parameter θ. We improve several circuit lower bounds for MCSP, using pseudorandom generators (PRGs) that are local; a PRG is called *local* if its output bit strings, when viewed as the truth table of a Boolean function, can be computed by a Boolean circuit of small size. We get new and improved lower bounds for MCSP that almost match the best-known lower bounds against several circuit models. Specifically, we show that computing MCSP, on functions with a truth table of length N, requires

- $N^{3-o(1)}$-size de Morgan formulas, improving the recent $N^{2-o(1)}$ lower bound by Hirahara and Santhanam (CCC, 2017),
- $N^{2-o(1)}$-size formulas over an arbitrary basis or general branching programs (no non-trivial lower bound was known for MCSP against these models), and
- $2^{\Omega\left(N^{1/(d+2.01)}\right)}$-size depth-$d$ AC^0 circuits, improving the superpolynomial lower bound by Allender et al. (SICOMP, 2006).

The AC^0 lower bound stated above matches the best-known AC^0 lower bound (for PARITY) up to a small *additive* constant in the depth. Also, for the special case of depth-2 circuits (i.e., CNFs or DNFs), we get an almost optimal lower bound of $2^{N^{1-o(1)}}$ for MCSP.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases minimum circuit size problem (MCSP), circuit lower bounds, pseudorandom generators (PRGs), local PRGs, de Morgan formulas, branching programs, constant depth circuits

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.39

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at https://eccc.weizmann.ac.il/report/2019/022/.

Acknowledgements We thank the anonymous ICALP'19 reviewers for their excellent comments.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 39; pp. 39:1–39:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Given the truth table of some Boolean function f and a size parameter θ, the minimum circuit size problem (MCSP) asks whether f can be computed by a circuit of size at most θ. Understanding the exact complexity of MCSP is an important open problem in computational complexity theory, dating back to the 1950s [20].

It is easy to see that MCSP is in NP. A popular conjecture is that MCSP is also NP-hard. However, despite serious efforts over the years, such a proof is still unknown. Given that it is difficult to show that MCSP is hard, perhaps the problem is easy? It turns out that this cannot be the case under some plausible cryptographic assumptions. More specifically, it is known that if one-way functions exist, then MCSP is not in P [11]. As proving an *unconditional* lower bound for MCSP seems far beyond the reach of currently known techniques, can we at least prove unconditional lower bounds for MCSP against some restricted computational models?[1]

Two of the most studied restricted computational models in complexity theory are constant-depth circuits (AC^0) and de Morgan formulas. For AC^0 circuits, the best-known lower bound is about PARITY: PARITY on N variables requires depth-d AC^0 circuits of size $2^{\Omega\left(N^{1/(d-1)}\right)}$ [6]. For de Morgan formulas, the state-of-the-art lower bound is almost cubic, namely $N^{3-o(1)}$, for some polynomial-time computable function [7, 18, 19, 5].

Notably, there are also lower bounds against these models for MCSP. Allender et al. [2] showed that MCSP, on functions represented as a truth table of length N, cannot be computed by polynomial-size constant-depth AC^0 circuits. In fact, by a more careful analysis of their argument, one can get a lower bound of $2^{N^{1/(c\cdot d+O(1))}}$, for a constant $c \geq 2$. However, such a lower bound still has a worse dependence on the depth compared to the PARITY lower bound. For de Morgan formulas, Hirahara and Santhanam [9] showed that computing MCSP requires de Morgan formulas of size $N^{2-o(1)}$.

Given these two MCSP lower bounds and the best-known lower bounds against these two models, it is natural to ask whether we can get MCSP lower bounds against small-depth circuits and de Morgan formulas that match the state-of-the-art lower bounds against these models. More specifically, can we show that computing MCSP requires depth-d AC^0 circuits of size $2^{N^{1/(d+O(1))}}$ and de Morgan formulas of size $N^{3-o(1)}$? Furthermore, can we show lower bounds for MCSP against some other restricted models that match their state-of-the-art lower bounds? In this paper, we answer these questions in the affirmative.

1.1 Our results

Our first result is an almost-cubic de Morgan formula lower bound for MCSP.

▶ **Theorem 1.** *Any de Morgan formula computing* MCSP *on truth tables of length N must have size at least $N^3/2^{O\left(\log^{2/3} N\right)}$.*

We also get almost-quadratic lower bounds against formulas over an arbitrary basis as well as general branching programs; these almost match the best-known lower bounds against these models [12].

▶ **Theorem 2.** *Let C be either a formula over any basis or a branching program that computes* MCSP *on truth tables of length N. Then C must have size at least $N^2/2^{O\left(\sqrt{\log N}\right)}$.*

[1] A recent line of research on *hardness magnification* [16, 14] provides another motivation for proving relatively weak lower bounds for restricted circuit models against certain "gap variants" of MCSP. Such lower bounds are shown to imply much stronger (superpolynomial) lower bounds.

For small-depth circuits, we have the following improved lower bound for MCSP, which its dependence on the depth matches the one in the PARITY lower bound, up to a small additive constant.

▶ **Theorem 3.** *For every $d > 2$ and every constant $\gamma > 0$, any depth-d AC^0 circuit computing* MCSP *on truth tables of length N must have size $2^{\Omega\left(N^{1/(d+2+\gamma)}\right)}$.*

For the special case of depth-2 circuits, we can have an almost optimal lower bound.

▶ **Theorem 4.** *Any CNF or DNF computing* MCSP *on truth tables of length N must have size $2^{N/\widetilde{O}\left(\log^2 N\right)}$.*

Also, in this paper (in the full version), we give a fine-grained analysis of the approach of obtaining MCSP lower bounds from average-case hardness via the Nisan-Wigderson framework.

1.2 Our techniques

For a class $\mathfrak{C}$ of N-variate Boolean functions, a pseudorandom generator (PRG) against $\mathfrak{C}$ is a deterministic efficiently-computable function G mapping short binary strings (seeds) to longer binary strings so that every function in $\mathfrak{C}$ accepts G's output on a uniformly random seed with about the same probability as that for an actual uniformly random string. A key notion in this work is that of a local PRG. We say that a PRG is *local* if its N-bit output (viewed as the truth table of some function) has small circuit complexity. More precisely, for any fixed seed to the PRG, there exists a small circuit such that, given $j \in [N]$ as an input, the circuit computes the j-th bit of the PRG output, where the size of the circuit is measured relative to its input length, namely $\log N$.

Local PRGs in the context of MCSP (and related problems) have been studied in previous works (see, e.g., [2, 15, 9, 8]). In this work, we refine the previous approaches, and obtain stronger circuit lower bounds by establishing strong locality properties of certain PRGs.[2]

MCSP lower bounds from local PRGs. Suppose we have a local PRG against some class of circuits $\mathfrak{C}$ of size s, and we want to show that MCSP cannot be computed by any size-s circuit in $\mathfrak{C}$. Suppose some size-s circuit C in $\mathfrak{C}$ computes MCSP. Using the fact that a random function has almost maximum circuit complexity, we have that C will output FALSE on most of its inputs (by setting the size parameter θ to be a non-trivial quantity that is asymptotically smaller than $2^n/n$, where n is the input length of the function). If we replace the uniformly random inputs with the outputs of the local PRG, then, by the definition of PRGs, C will still output FALSE with large probability. However, since the PRG is local, all of its outputs have circuit complexity smaller than the size parameter θ, and hence must be accepted by C. A contradiction.

To get a strong lower bound, we would like to make the above argument to work for large s. Note that the local complexity of the PRG, $\lambda(s)$, is a function on the size of the circuit C, and we need this local complexity to be "non-trivial" in order to reach a contradiction. Therefore, we want to choose s so that this local complexity remains asymptotically smaller than $2^n/n$. As a result, the final lower bound (i.e., the largest s that we can choose) is determined by the local complexity λ. So the main question we study in our paper is: What is the smallest local complexity of a PRG against a given circuit class?

[2] Note that, as one of our reviewers pointed out, the notion of a local PRG can be also found in the context of cryptography [4], where a PRG $G : \{0,1\}^n \to \{0,1\}^m$ is called *k-local*, for some constant $k > 0$, if every output PRG bit $G(x)_j$, for any $x \in \{0,1\}^n$ and $j \in [m]$, depends only on k input bits $x_{i_1}, \ldots, x_{i_k}$, for $i_1, \ldots, i_k \in [n]$. In our work, however, locality refers to the circuit complexity of the PRG at hand and the output bits of our PRGs may depend on a superconstant number of input bits.

MCSP lower bound against de Morgan formulas. Our formula lower bound for MCSP is obtained by applying the framework described above to a local PRG against formulas. The state-of-the-art PRG against formulas is given by Impagliazzo, Meka, and Zuckerman [10], which we refer to as the IMZ PRG. Their PRG has a seed length of $s^{1/3+o(1)}$ for size s formulas (note that such a PRG is useful against sub-cubic formulas only). If we want to utilize the IMZ PRG to get an MCSP lower bound against formulas, we will need to argue that the IMZ PRG is local.

In fact, in order to get an almost-cubic lower bound, we will need such a PRG to be strongly local in the sense that any single output bit of the PRG (on any given fixed seed) can be computed by a circuit of size comparable to its seed length, which is $s^{1/3+o(1)}$. However, by inspecting the construction, the IMZ PRG does not seem to have such a property, and a straightforward implementation seems to require a circuit of size at least $s^{2/3}$ (see the full version for more details), which yields a weaker lower bound for MCSP.

To overcome this issue, we present an alternative PRG useful against sub-cubic formulas which is strongly local. The construction of this PRG can be viewed as a modification of the IMZ PRG. At a high level, it is based on the Ajtai-Wigderson construction [1], which is a framework for constructing PRGs against computations that can be simplified under (pseudo)random restrictions. This framework is then combined with the ideas of reducing (recycling) random bits using an extractor, by exploiting communication bottlenecks in computations [13]. Our modification, particularly the utilization of the Ajtai-Wigderson construction, allows us to compute any output bit of the PRG efficiently by reducing the number of calls to the extractor. Using some crucial observations on the circuit complexity of certain pseudorandom objects, we get a PRG that is locally computable by a $s^{1/3+o(1)}$-size circuit.[3]

MCSP lower bounds against formulas over an arbitrary basis or branching programs. The MCSP lower bounds against formulas over an arbitrary basis or branching programs are obtained similarly to those for de Morgan formulas. The idea is to construct strongly local PRGs against these models by modifying the PRGs in [10]. Then, by applying our "MCSP circuit lower bounds from local PRGs" framework, we get the desired lower bounds.

MCSP lower bounds against AC^0. We use a local PRG against AC^0 to get MCSP lower bounds. To get a lower bound matching the one in Theorem 3, we can use the state-of-the-art PRG against AC^0 by Trevisan and Xue [21], which has a seed length of $(\log s)^{d+O(1)}$ for size-s depth-d AC^0 circuits. By a careful analysis of the construction of this PRG, we can show that the Trevisan-Xue PRG is strongly local and can be used to get an MCSP lower bound that is close to the one stated in Theorem 3. However, in this paper, we will present a more direct proof of such a lower bound by using the pseudorandom switching lemma for constant-depth circuits, which is due to Trevisan and Xue [21], as well, and is a key ingredient in their PRG.

The idea is to show that for any small-depth circuit of size less than the claimed lower bound, there is some locally computable restriction that turns the circuit into a constant function, but leaves many variables unrestricted. However, MCSP cannot be constant under such a restriction, because depending on the partial assignment to the unrestricted variables, the resulting input function (which is composed of the restriction and the partial assignment) can be either easy or hard. Such an approach based on pseudorandom restrictions can also be applied to depth-2 circuits and yield almost optimal CNF (and DNF) MCSP lower bounds.

[3] It is also possible to use the original IMZ PRG to obtain an almost-cubic formula lower bound for MCSP. We can show that the IMZ PRG, although not fully strongly local, is "almost strongly local" in the sense that *most* of its outputs have very small circuit complexity; see the full version.

1.3 Remainder of the paper

We give the necessary background in Section 2. In Section 3, we describe our framework of using local PRGs to obtain lower bounds for MCSP. We prove the almost-cubic de Morgan formula lower bound for MCSP (Theorem 1) in Section 4, and the almost-quadratic lower bound against formulas over an arbitrary basis and branching programs (Theorem 2) in Section 5. The improved AC^0 lower bounds for MCSP (Theorem 3 and Theorem 4) are proved in Section 6. Finally, we give some open problems in Section 7. Due to space limitations we relegated some material to the full version, like some omitted proofs and the framework of proving MCSP lower bounds from average-case hardness.

2 Preliminaries

2.1 Notation

For any computational model, we use the term *size* to refer to its complexity measure. For example, if the model is circuits of some fixed depth, then the size is the number of gates in the circuit.

For a positive integer n, that is a power of two, we use the following notation: $[n]$ denotes the set $\{1, 2, \ldots, n\} \cong \{0,1\}^{\log n}$, $\mathbb{F}_n$ denotes the field with n elements, where the elements in $\mathbb{F}_n$ are represented by $(\log n)$-bit strings, U_n denotes the uniform distribution over $\{0,1\}^n$, and, for a function $f\colon \{0,1\}^n \to \{0,1\}$, $\mathtt{tt}(f) \in \{0,1\}^{N=2^n}$ denotes the truth table of f.

2.2 Pseudorandomness

▶ **Definition 5** (Pseudorandom generators). *Let $G\colon \{0,1\}^r \to \{0,1\}^n$ be a function, $\mathcal{F}$ be a class of Boolean functions, and $0 < \varepsilon < 1$. We say that G is a* pseudorandom generator of seed length r that ε-fools $\mathcal{F}$ *if, for every function $f \in \mathcal{F}$, it is the case that*

$$\left|\mathbb{E}_{z\sim\{0,1\}^r}[f(G(z))] - \mathbb{E}_{x\sim\{0,1\}^n}[f(x)]\right| \leq \varepsilon.$$

▶ **Definition 6** (k-wise independence). *A distribution X over $[m]^n$ is called* k-wise independent *if for any $1 \leq i_1 \leq i_2 \leq \cdots \leq i_k \leq n$ and every $b_1, b_2, \ldots, b_k \in [m]$, we have*

$$\mathbf{Pr}[X_{i_1} = b_1, X_{i_2} = b_2, \ldots, X_{i_k} = b_k] = m^{-k}.$$

The following simple fact (proved in the full version) will be convenient for us.

▶ **Lemma 7.** *Let X and Y be two random variables that take values in $\{0,1\}$ and $\mathcal{E}$ be some event. If $|\mathbb{E}[X \mid \mathcal{E}] - \mathbb{E}[Y \mid \mathcal{E}]| \leq \varepsilon_1$ and $\mathbf{Pr}[\neg\mathcal{E}] \leq \varepsilon_2$, then $|\mathbb{E}[X] - \mathbb{E}[Y]| \leq \varepsilon_1 + \varepsilon_2$.*

2.3 Random restrictions

A restriction for a n-variate Boolean function f, usually denoted as $\rho \in \{0,1,*\}^n$, specifies a way of fixing the values of some subset of variables for f. We denote by f_ρ the restricted function after the variables are restricted according to ρ, and denote by $\rho^{-1}(*)$ the set of unrestricted variables. A random restriction is then a distribution over restrictions, which can be specified by a pair $(\sigma, \beta) \in \{0,1\}^n \times \{0,1\}^n$, where σ (as a characteristic string) specifies the set of unrestricted variables, and β specifies the values for fixing the restricted variables. We say that a random restriction (or random selection) is p-regular if each variable is left unrestricted with probability p. One way to generate a p-regular random restriction is to leave each variable, independently, unrestricted with probability p, and otherwise assign to it a 0 or

a 1, uniformly at random. Such a random restriction is called a *(truly) p-random restriction*. Note that to sample such a restriction, we can first pick a string in $\{0,1\}^{n\cdot\log(1/p)} \cong [1/p]^n$ to specify the selection of the unrestricted variables, where a coordinate is unrestricted if and only if all of its corresponding $\log(1/p)$ bits are 0, and then a string in $\{0,1\}^n$ to specify the values assigned to each of the restricted variables. So sampling a restriction in this way requires $n \cdot \log(1/p) + n$ random bits.

We can also generate a restriction in a pseudorandom manner, which may use fewer random bits. For example, one way to do this is to use a limited-independence distribution over $[1/p]^n$, so that each variable is left unrestricted with probability p and any k variables are independent. Also, we can let each variable be assigned a 0 or a 1, uniformly at random, in a way such that any k of the variables are independent; this again can be done using a k-wise independent distribution on $\{0,1\}^n$.

2.4 Simple facts about Boolean circuits

▶ **Proposition 8.** *A Boolean circuit of size s can be specified using $O(s \log s)$ bits. Hence there are at most $2^{O(s \log s)} = s^{O(s)}$ distinct circuits of size at most s.*

▶ **Theorem 9** ([17])**.** *The fraction of functions on n variables that have a circuit of size less than $2^n/(3n)$ is $o(1)$.*

The following lemma is proved in the full version.

▶ **Lemma 10.** *For any integer $t > 0$, there exists a circuit C of size $\widetilde{O}(t)$ such that, given any string $x \in \{0,1\}^t$, the circuit does the following: If $x = 0^t$, then C outputs $(0, 0^{\log t})$ and if $x \neq 0^t$, then C outputs $(1, q)$, where $q \in \{0,1\}^{\log t}$ is the index of the first bit in x that is not 0.*

The following circuit upper bound for the addressing (storage access) function is well-known (see, e.g., [22]); we include a proof, in the full version, for completeness.

▶ **Lemma 11.** *For any integers $t, m > 0$, there exists a circuit of size $O(tm)$ such that, given any string $y = (y_1, y_2, \ldots, y_t)$, where $y_i \in \{0,1\}^m$ for each i, and an index $i \in \{0,1\}^{\log t}$, the circuit outputs y_i.*

3 The "MCSP circuit lower bounds from local PRGs" framework

We first describe how to use local PRGs to obtain MCSP lower bounds.

▶ **Definition 12** (Local PRGs)**.** *Let $\lambda\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a size function. For any Boolean computational model and size $s > 0$, we say that a function $G\colon \{0,1\}^{r=r(N,s)} \to \{0,1\}^N$ is a $(N, s, \lambda(N,s))$-local PRG against the model if G 1/3-fools every device f on N variables of size s in the model; that is,*

$$\left|\mathbb{E}_{z\sim\{0,1\}^r}[f(G(z))] - \mathbb{E}_{x\sim\{0,1\}^N}[f(x)]\right| \leq 1/3,$$

and for any seed $z \in \{0,1\}^r$, the function $g\colon \{0,1\}^{\log N} \to \{0,1\}$, defined as $g_z(j) = G(z)_j$, can be computed by a general circuit of size at most $\lambda(N,s)$.

Note that $\lambda(N,s)$ is at least $r(N,s)$, by a counting argument (neglecting $\log \lambda(N,s)$ factors). This is to ensure that, for any function g, on n variables, which may be output by G, there is some $\lambda(N,s)$-size circuit that computes g.

▶ **Theorem 13.** *There exists a constant $c > 0$ such that the following holds. For any computational model, let s be such that* MCSP *on truth tables of length N can be computed by a device of size s in the model. If there exists some $(N, s, \lambda(N, s))$-local PRG against the model, then $\lambda(N, s) \geq \frac{N}{c \log N}$.*

Proof. Let C be a device in the computational model such that C computes MCSP on truth tables of length N. Suppose C has size s, and let G be a $(N, s, \lambda(N, s))$-local PRG against C with some seed length r.

For the sake of contradiction, suppose that $\lambda(N, s) < \frac{N}{c \log N}$. On the one hand, since most functions require circuits of size greater than $\frac{N}{c \log N}$ (Theorem 9) and C computes MCSP, we have $\mu = \mathbf{Pr}_{\mathtt{tt}(f) \sim \{0,1\}^N}[C(\mathtt{tt}(f), \lambda(N, s)) = 0] \geq 1/2$. Also, since G fools C, we have $\mathbf{Pr}_{z \sim \{0,1\}^r}[C(G(z), \lambda(N, s)) = 0] \geq \mu - 1/3 \geq 1/6$. On the other hand, because G is $(N, s, \lambda(N, s))$-local, we must have $C(G(z), \lambda(N, s)) = 1$, for every z. A contradiction. ◀

4 Almost-cubic de Morgan formula lower bounds for MCSP

In this section, we present our almost-cubic de Morgan formula lower bound for MCSP. By saying "formula" within this section, we refer to formulas over the de Morgan basis (AND, OR, and NOT). By *size* of a formula, we mean its usual leaf complexity, i.e., the number of leaves in the tree representation of the formula.

▶ **Theorem 14** (Theorem 1 restated)**.** *Any de Morgan formula computing* MCSP *on truth tables of length N must have size at least $N^3/2^{O\left(\log^{2/3} N\right)}$.*

We will construct a strongly local PRG useful against sub-cubic formulas. That is, given as input an index j, the j-th bit of the PRG can be computed by a circuit of size that is comparable to its seed length, which in our case is around $s^{1/3}$ for size s formulas.

▶ **Lemma 15.** *For any $s \geq N$, there exists a $\left(N, s, s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}\right)$-local PRG against de Morgan formulas.*

Given the local PRG in Lemma 15, we can combine it with our Theorem 13 to obtain a formula lower bound for MCSP.

Proof of Theorem 14. Let s be such that MCSP on truth tables of length N can be computed by some formula of size s. We can assume that $s \leq N^3$ since, otherwise, the result trivially holds. By Theorem 13 and Lemma 15, we have $s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)} \geq N/(c \log N)$; then, $s \geq N^3 / \left(2^{O\left(\log^{2/3} N\right)} c^3 \log^3 N\right)$. ◀

The rest of this section is devoted to proving Lemma 15.

4.1 Almost-linear-size k-independent generators

The PRG in Lemma 15 will use k-wise independent distributions. Recall that a multidimensional distribution is called *k-wise independent* if any k coordinates of the distribution are uniformly distributed (see Definition 6). We say that a function G is a *k-independent generator* if, for random inputs, the distribution of the outputs of G is k-wise independent.

We will need a k-independent generator that is strongly local.

▶ **Lemma 16.** *For any integer $k > 0$, there exists a k-independent generator $G\colon \{0,1\}^r \to [m]^N$, with $r = k \cdot \max\{\log N, \log m\}$, such that the following holds. There exists a circuit of size $k \cdot \max\{\widetilde{O}(\log N), \widetilde{O}(\log m)\}$ such that, given $j \in \{0,1\}^{\log N}$ and a seed $z \in \{0,1\}^r$, the circuit computes the j-th coordinate of $G(z)$ (as an element of $\{0,1\}^{\log m}$).*

The above k-independent generator is constructed using finite fields (see the full version). Its efficiency crucially depends on the fact that finite field arithmetic can be done using *almost linear-size* Boolean circuits.

4.2 Almost-linear-size extractors

Our PRG will make use of randomness extractors. Here, we describe an extractor that is computable by a circuit of size that is almost linear in the length of its input. We start by reviewing the definitions of some basic notions regarding extractors.

▶ **Definition 17** (ε-closeness and statistical distance). *Let $0 \leq \varepsilon \leq 1$. We say two distributions X and Y (over some universe D) are* ε-close *if their* statistical distance, *defined as*

$$\max_{T:D\to\{0,1\}} \left|\mathbf{Pr}[T(X)=1] - \mathbf{Pr}[T(Y)=1]\right|,$$

is at most ε.

▶ **Definition 18** (Min-entropy). *Let X be a random variable. The* min-entropy *of X, denoted by $H_\infty(X)$, is the largest real number k such that $\mathbf{Pr}[X=x] \leq 2^{-k}$ for every x in the range of X. If X is a distribution over $\{0,1\}^\aleph$ with $H_\infty(X) \geq k$, then X is called a* $(\aleph, k)$-source.

▶ **Definition 19** (Extractors). *A function $E\colon \{0,1\}^\aleph \times \{0,1\}^d \to \{0,1\}^m$ is an* (k,ε)-extractor *if, for any $(\aleph,k)$-source X, the distribution $E(X,U_d)$ is ε-close to U_m.*

We now state the extractor, which for a high min-entropy source extracts a constant fraction of min-entropy, using seeds of polylogarithmic length. The construction and circuit complexity of this extractor are presented in the full version.

▶ **Lemma 20** (Almost-linear-size extractors, following [13]). *There exists some randomness extractor $E\colon \{0,1\}^\aleph \times \{0,1\}^d \to \{0,1\}^m$ that is an $(\aleph/2,\varepsilon)$-extractor with $m = \Omega(\aleph)$ and $d = \mathsf{polylog}(\aleph/\varepsilon)$. Moreover, E can be computed by a circuit of size $\aleph \cdot \mathsf{polylog}(\aleph/\varepsilon)$.*

4.3 Strongly local PRG useful against sub-cubic de Morgan formulas

For a formula F, let $L(F)$ denote the size (which is measured by the number of leaves) of F. We need the following pseudorandom shrinkage lemma for de Morgan formulas, which says that there exists a p-regular restriction, where the unrestricted variables are selected pseudorandomly and the restricted variables are fixed truly-randomly, such that *with high probability* the size of the restricted formula will "shrink" by a factor of p^2.

▶ **Lemma 21** (Pseudorandom shrinkage lemma, Lemma 4.8 of [10][4]). *There exists a constant $c_0 > 0$ such that the following holds. For any constant $c > c_0$, any $s \geq N$, $p \geq s^{-1/2}$, and any de Morgan formula F on N variables of size s, there exists a p-regular pseudorandom selection $\mathcal{D}$ over N variables, that is samplable using $r = 2^{O(\log^{2/3} s)}$ random bits, such that*

$$\mathbf{Pr}_{\sigma\sim\mathcal{D}, x\sim\{0,1\}^N}\left[L\big(F_{(\sigma,x)}\big) \geq 2^{3\cdot c\cdot \log^{2/3} s} \cdot p^2 \cdot s\right] \leq s^{-c}.$$

Moreover, there exists a circuit of size $2^{O(\log^{2/3} s)}$ such that, given $j \in \{0,1\}^{\log N}$ and a seed $z \in \{0,1\}^r$, the circuit computes the j-th bit of $\mathcal{D}(z)$.

[4] The pseudorandom shrinkage lemma in [10] is not stated in this form, but rather selects the unrestricted variables and fixes the restricted variables both pseudorandomly (based on limited independence). However, our version here follows from the proof of the original version in Section 4.2 of [10] by noting that limited-independence distributions can be computed locally.

We are now ready to show our PRG in Lemma 15.

Proof of Lemma 15. The construction is as follows: We first sample a p-regular pseudorandom selection from Lemma 21. Then, we fill the star coordinates, specified by the pseudorandom selection, in the output string with the output of some extractor which takes a min-entropy source sample and a short seed (in fact, it is the output of some limited-independence generator that takes the output of the extractor as a seed). We then sample another pseudorandom selection, and fill the star coordinates specified by this pseudorandom selection but this time only for those that have not been filled in previous steps, again with the output of the same extractor using the *same min-entropy source sample* but a *different short seed*. We continue this way until all the coordinates are filled.

More formally, our PRG uses the following parameters:[5]

- $p = 1/s^{1/3}$, the expected fraction of unrestricted variables in each of the pseudorandom selections;
- $\varepsilon = 1/\mathsf{poly}(N)$ and $\varepsilon_0 = \varepsilon/(10t)$, which specify the error of the PRG;
- $t = \ln(2N/\varepsilon)/p = s^{1/3} \cdot O(\log N)$, the number of steps needed so that all the coordinates will be filled with probability except $\varepsilon/2$;
- $s_0 = p^2 \cdot s \cdot 2^{O\left(\log^{2/3} s\right)} = s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$, the size of the formula after being simplified by a pseudorandom restriction;
- $k \geq s_0 = s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$, the amount of independence needed to fool the simplified formula, and $r_k = k \cdot \log N$ the seed length for the k-independent generator;
- $\aleph$, the length of the min-entropy source for the extractor, which is such that $\aleph \geq 2 \cdot (\log(1/\varepsilon_0) + c \cdot s_0 \cdot \log s_0)$, where $c > 0$ is some constant, and that $\aleph = \Omega(r_k)$. We can take $\aleph = s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$;
- $d = \mathsf{polylog}(\aleph/\varepsilon_0) = \mathsf{polylog}(N)$, the seed length of the extractor;
- $\ell = 2^{O(\log^{2/3} s)}$, the number of random bits for sampling a pseudorandom selection.

Construction. The PRG takes a seed $(X, Y_1, Y_2, \ldots, Y_t, \gamma_1, \gamma_2, \ldots, \gamma_t) \in \{0,1\}^r$, where

- $X \in \{0,1\}^{\aleph}$ is the min-entropy source sample of an extractor,
- $Y_i \in \{0,1\}^{\mathsf{polylog}(N)}$, for each $i \in [t]$, is the seed of an extractor, and
- $\gamma_i \in \{0,1\}^{\ell}$, for each $i \in [t]$, is the seed for sampling a pseudorandom selection.

The construction of the PRG proceeds in the following two stages.

1. Compute a sequence of t p-regular pseudorandom selections $\sigma_1, \ldots, \sigma_t$, using Lemma 21, with the seeds $\gamma_1, \ldots, \gamma_t$. Below, we denote the star coordinates in σ_i by $\sigma_i^{-1}(*)$. Let $S_1, \ldots, S_t \subseteq [N]$ be t disjoint sets defined by

$$S_i = \sigma_i^{-1}(*) \setminus (S_1 \cup S_2 \cup \cdots \cup S_{i-1}).$$

2. Define $Z_1, Z_2, \ldots, Z_t \in \{0,1\}^N$ by

$$Z_i = G_k(E(X, Y_i)),$$

where $E\colon \{0,1\}^{\aleph} \times \{0,1\}^d \to \{0,1\}^{\Omega(\aleph)}$ is an $(\aleph/2, \varepsilon_0)$-extractor and $G_k\colon \{0,1\}^{r_k} \to \{0,1\}^N$ is a k-independent generator. The final output of our PRG is the binary string that has the values $Z_i|_{S_i}$ in the positions indexed by S_i, for all $i \in [t]$, where $Z_i|_{S_i}$ denotes the bit values of Z_i projected to the set S_i. (We fix those positions that are not in any of the S_i's to be 0.) Stage 2 of the PRG construction is depicted in Figure 1.

[5] In fact, there are mainly two types of parameters here. Those that are close to $s^{1/3}$, which are $1/p, t, s_0, k, N$, and those that are close to $N^{o(1)}$, which are d and ℓ.

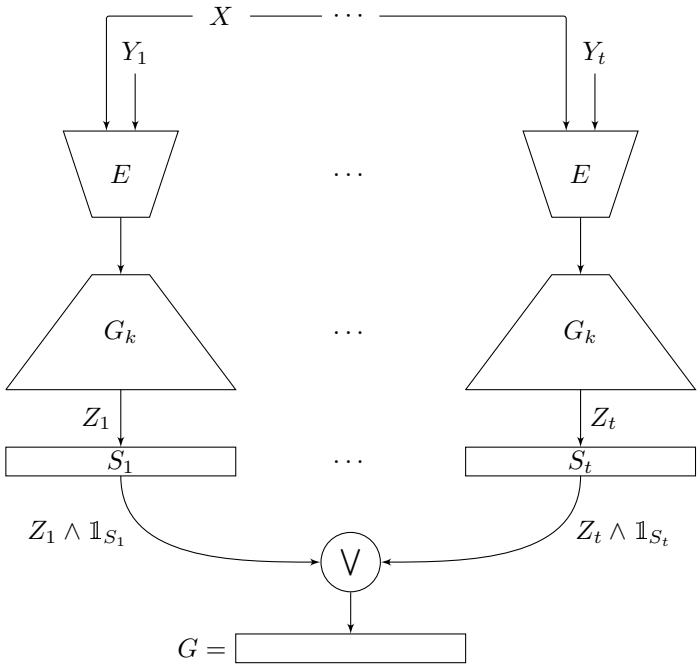

Figure 1 Construction of the PRG in Lemma 15, Stage 2. For each $i \in [t]$, $\mathbb{1}_{S_i} \in \{0,1\}^N$ denotes the characteristic Boolean vector of the set S_i, where $S_i \subseteq [N]$ is the set of star coordinates in the i-th pseudorandom selection that did not appear in the preceding sets $S_1, \ldots, S_{i-1}$. Also, $\wedge$ denotes a coordinate-wise AND operation (i.e., coordinate-wise *multiplication* of Boolean vectors) and $\bigvee$ is a coordinate-wise OR operation.

Correctness. Next, we show that the above PRG ε-fools N-variate formulas f of size s. First, note that, by our choice of t, with probability except $\varepsilon/2$, $S_1 \cup S_2 \cup \cdots \cup S_t$ covers all coordinates. For the rest of the argument, we will assume that this is the case. By Lemma 7, conditioning on this assumption contributes at most $\varepsilon/2$ to the error of the constructed PRG.

We continue the correctness analysis using a *hybrid argument*. Let G denote the distribution given by the PRG described above. Let U be the uniform distribution. Note that if in the above construction we replace Z_i, for all $i \in [t]$, with U, then we would get a uniform distribution. Now we can start from there and gradually replace U with the Z_i's step-by-step for a total of t steps. We will argue that after each replacement step, the expected value of the function does not change by much. Let A_i be the distribution so that we have replaced U with Z_i in the first i steps. That is,

$$A_i = \left(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, U|_{S_{i+1}}, \ldots, U|_{S_t} \right) = \left(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, U|_{S_{i+1} \cup \cdots \cup S_t} \right).$$

For the sake of contradiction, suppose there exists an N-variate size-s formula f such that

$$|\mathbb{E}[f(U)] - \mathbb{E}[f(G)]| = |\mathbb{E}[f(A_0)] - \mathbb{E}[f(A_t)]| > \varepsilon/2.$$

By the triangle inequality, there exists an $0 \leq i < t$ such that

$$|\mathbb{E}[f(A_i)] - \mathbb{E}[f(A_{i+1})]| > \varepsilon/(2t). \tag{1}$$

Let us say that the expectations in Equation (1) are over $\sigma_1, \ldots, \sigma_{i+1}, Y_1, \ldots, Y_{i+1}, X, U$, and we may remove the absolute value without loss of generality. Then, we have

$$\mathbb{E}_{\substack{\sigma_1,\ldots,\sigma_i,\\ Y_1,\ldots,Y_i,\\ X}} \left[\mathbb{E}_{\sigma_{i+1},Y_{i+1},U} \left[f(A_i) \right] - \mathbb{E}_{\sigma_{i+1},Y_{i+1},U} \left[f(A_{i+1}) \right] \right] > \varepsilon/(2t). \tag{2}$$

Let $W_i = (\sigma_1, \ldots, \sigma_i, Y_1, \ldots, Y_i, X)$, and let f' be the random function (where the randomness is over W_i) defined as $f' = f(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, \quad \cdot \quad)$. That is, f' is the restricted function after the first i steps. Then, the left hand side of Equation (2) becomes

$$\mathbb{E}_{W_i}\left[\mathbb{E}_{\sigma_{i+1},U}\left[f'(U|_{S_{i+1}}, U|_{\ S_{i+2}\cup\cdots\cup S_t})\right] - \mathbb{E}_{\sigma_{i+1},Y_{i+1},U}\left[f'(Z_{i+1}|_{S_{i+1}}, U|_{\ S_{i+2}\cup\cdots\cup S_t})\right]\right]. \tag{3}$$

Note that, at this point, we can view $\rho_{i+1} = (\sigma_{i+1}, U)$ as a pseudorandom restriction (in the sense of Lemma 21) applied to f'. Next, let f'' be the random function defined as the restricted function of f' under ρ_{i+1} (note that the randomness is over W_i, and also the pseudorandom restriction ρ_{i+1}). Now Equation (3) becomes

$$\mathbb{E}_{W_i,\rho_{i+1}}\left[\mathbb{E}_U\left[f''(U)\right] - \mathbb{E}_{Y_{i+1}}\left[f''(Z_{i+1})\right]\right]. \tag{4}$$

Note that in the above, we abuse notation and use U and Z_{i+1} to denote $U|_{S_{i+1}}$ and $Z_{i+1}|_{S_{i+1}}$, respectively.

Next we want to show that the difference between the two expectations in Equation (4) is at most $3\varepsilon_0 = 3\varepsilon/(10t) \leq \varepsilon/(2t)$, which would give a contradiction, by Equation (2). The intuition is the following. On the one hand, f'' is obtained by a pseudorandom restriction ρ_{i+1}, and so, with high probability, it has size at most s_0. On the other hand, Z_{i+1} is obtained using an extractor that is supposed to extract enough random bits for an s_0-independent generator.

The issue, however, is that f'' depends on X, the source sample of the extractor. Therefore, f'' may contain information about X, so that X is not truly random anymore. Nonetheless, being a formula of size at most s_0, f'' cannot contain too much information, and so cannot take too much entropy away from X. We make this argument more formal next.

Let us define the set of good functions for f'', namely

$$\mathcal{F} = \left\{g \mid L(g) \leq s_0 \quad \text{and} \quad \mathbf{Pr}_{W_i,\rho_{i+1}}[f'' = g] \geq \varepsilon_0/s_0^{cs_0}\right\},$$

where c is some constant. Let $\mathcal{E}$ denote the event $f'' \in \mathcal{F}$. We first show the following.

▷ Claim 22. It is the case that $\mathbf{Pr}[\neg\mathcal{E}] \leq 2\varepsilon_0$.

Proof of Claim 22. We have

$$\begin{aligned}\mathbf{Pr}[\neg\mathcal{E}] &= \mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') > s_0)] + \mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') \leq s_0)] \\ &\leq \mathbf{Pr}[(L(f'') > s_0)] + \mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') \leq s_0)].\end{aligned}$$

Note that, by the pseudorandom shrinkage lemma (Lemma 21), we have $\mathbf{Pr}[L(f'') > s_0] \leq \varepsilon_0$. Also note that under the condition that $L(f'') \leq s_0$, there can be at most $s_0^{O(s_0)}$ choices for f'', since a formula of size s_0 can be specified using $O(s_0 \log s_0)$ bits (Proposition 8). Therefore, $\mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') \leq s_0)] \leq s_0^{O(s_0)} \cdot \varepsilon_0/s_0^{cs_0} \leq \varepsilon_0$. ◁

Let us now analyze Equation (4) while conditioning on the event $\mathcal{E}$. We show the following.

▷ Claim 23. It is the case that $\mathbb{E}\left[f''(U) \mid \mathcal{E}\right] - \mathbb{E}\left[f''(Z_{i+1}) \mid \mathcal{E}\right] \leq \varepsilon_0$.

Proof of Claim 23. First note that conditioning on $\mathcal{E}$, X still has large min-entropy. More precisely, for every $g \in \mathcal{F}$ it is the case that $H_\infty(X \mid f'' = g) \geq \aleph/2$. This is because, for every x in the range of X, we have

$$\mathbf{Pr}[X = x \mid f'' = g] \leq \frac{\mathbf{Pr}[X = x]}{\mathbf{Pr}[f'' = g]} \leq \frac{2^{-\aleph}}{\varepsilon_0/s_0^{c\cdot s_0}} = 2^{-(\aleph - \log(1/\varepsilon_0) - c\cdot s_0 \cdot \log s_0)} \leq 2^{-\aleph/2}.$$

Then, by the definition of the extractor, we have $\mathbb{E}\left[f''(G_k(U)) \mid \mathcal{E}\right] - \mathbb{E}\left[f''(Z_{i+1}) \mid \mathcal{E}\right] \leq \varepsilon_0$. Finally, note that $\mathbb{E}\left[f''(G_k(U)) \mid \mathcal{E}\right] = \mathbb{E}\left[f''(U) \mid \mathcal{E}\right]$, since s_0-wise independent distributions fool size-s_0 formulas. ◁

Combining Claim 22, Claim 23, and Lemma 7, we get that the quantity in Equation (4) is at most $3\varepsilon_0$, which leads to a contradiction. This completes the proof of correctness.

Locality. To see that the j-th bit of the PRG can be computed using a circuit of size $s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$, we observe the following equivalent construction:

1. Compute the j-th bits of the t pseudorandom selections $(\sigma_1)_j, (\sigma_2)_j, \ldots, (\sigma_t)_j$.
2. Retrieve Y_q, where q is the smallest integer such that $(\sigma_q)_j$ is a star.
3. Compute $(Z_q)_j = G_k(E(X, Y_q))_j$, as the j-th bit of the PRG.

Note that Step 1 can be done using a circuit of size $t \cdot 2^{O\left(\log^{2/3} s\right)} = s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$, by the pseudorandom shrinkage lemma (Lemma 21). Also, Step 2 can be done by first computing q from the sequence $((\sigma_i)_j)_{i\in[t]}$, using a circuit of size $\widetilde{O}(t)$ (Lemma 10), and then outputting Y_q from $(Y_i)_{i\in[t]}$, using a circuit of size $t \cdot \mathsf{polylog}(N)$ (Lemma 11). Finally, Step 3 can be done by a circuit of size $\widetilde{O}(\aleph)$, using the efficient extractor (Lemma 20) and the limited-independence generator (Lemma 16). ◀

5 Almost-quadratic lower bounds against arbitrary basis formulas and branching programs

The MCSP lower bounds against formulas, over an arbitrary basis, and branching programs are obtained similarly to those for de Morgan formulas in the previous section. The idea is to construct strongly local PRGs against these models by modifying the PRGs in [10].

▶ **Lemma 24.** *For any $s \geq n$, there exists a $\left(N, s, s^{1/2} \cdot 2^{O\left(\sqrt{\log s}\right)}\right)$-local PRG against size-$s$ formulas over an arbitrary basis (or branching programs).*

The MCSP lower bound in Theorem 2 follows from Lemma 24 and Theorem 13.

6 Improved AC^0 lower bounds for MCSP

In this section, we show improved lower bounds for MCSP against constant-depth circuits.

6.1 The case of depth $d > 2$

We first show the improved lower bound against circuits of depth $d > 2$ that almost matches the lower bound for PARITY.

▶ **Theorem 25** (Theorem 3 restated). *For every $d > 2$ and every constant $\gamma > 0$, any depth-d AC^0 circuit computing MCSP on truth tables of length N must have size $2^{\Omega\left(N^{1/(d+2+\gamma)}\right)}$.*

The above result is proved using the following structural property of small-depth circuits, which says that for any such circuit, there exists some locally computable restriction that simplifies the circuits to be a constant while leaving many variables unrestricted.

▶ **Lemma 26.** *For any size-s depth-d circuit C, there exists a restriction $\rho \in \{0,1,*\}^N$ such that C_ρ is a constant function, $\left|\rho^{-1}(*)\right| \geq \frac{N}{O(\log s)^{d-2}} - \log s$, and there exists a circuit of size $d \cdot \log(N) \cdot \widetilde{O}\left(\log^3 s\right)$ such that, given $j \in \{0,1\}^{\log N}$, the circuit computes the j-th coordinate of ρ.*

The proof of Lemma 26, presented in the full version, uses the pseudorandom switching lemma due to Trevisan and Xue [21], which says that a depth-2 circuit is likely to be simplified after hit by a pseudorandom restriction. We now prove Theorem 25.

Proof of Theorem 25. Let C be a depth-d AC^0 circuit on $\{0,1\}^N \times \{0,1\}^{\log N}$ such that C computes MCSP on truth tables of length N, and let s be the size of C.

For a size parameter $\lambda = d \cdot \log(N) \cdot \widetilde{O}\left(\log^3 s\right)$, let $C' = C(\cdot, \lambda)$. Let ρ be a restriction from Lemma 26 for C'. By Lemma 26, we have that C'_ρ is a constant function. First note that $C'_\rho(0^{|\rho^{-1}(*)|}) = 1$. To see this, note that $C'_\rho(0^{|\rho^{-1}(*)|}) = C(\mathtt{tt}(f), \lambda)$, where C computes MCSP and $f\colon \{0,1\}^{\log N} \to \{0,1\}$ is the following:

$$f(j) = \begin{cases} 0, & \text{if } \rho_j = 0 \text{ or } \rho_j = *, \\ 1, & \text{else if } \rho_j = 1. \end{cases}$$

By Item 3 of Lemma 26, such a function f can be computed by a λ-size circuit. On the other hand, there can be $2^{|\rho^{-1}(*)|}$ different functions corresponding to the different partial assignments to the unrestricted variables. Since there are at most $O(\lambda \log \lambda)$ different circuits of size at most λ, in order for C'_ρ to be the constant 1, we must have $2^{O(\lambda \log \lambda)} \geq 2^{|\rho^{-1}(*)|} = 2^{\frac{N}{O(\log s)^{d-2}} - \log s}$, which, by a simple calculation, implies $s = 2^{\Omega\left(N^{1/(d+2+\gamma)}\right)}$, for any $\gamma > 0$. ◀

6.2 The case of depth 2

Here we show that computing MCSP requires depth-2 circuits of almost maximum size.

▶ **Theorem 27** (Theorem 4 restated). *Any CNF or DNF computing* MCSP *on truth tables of length N must have size $2^{N/\widetilde{O}\left(\log^2 N\right)}$.*

The proof uses a variant of Lemma 26 which says that a depth-2 circuit can be made constant via a more efficient restriction. Given such a local restriction, it is straightforward to prove Theorem 27 following the argument in the proof of Theorem 25.

7 Open problems

Our de Morgan formula lower bound for MCSP is still slightly weaker than the state-of-the-art de Morgan formula lower bound due to Tal [19], which is $\Omega\left(N^3/\left(\log N \cdot (\log\log N)^2\right)\right)$. Can the MCSP lower bound be improved? Are there better constructions of local PRGs against formulas? Or, are there alternative proofs that do not rely on local PRGs?

A similar question can be asked for small-depth circuits. In particular, can we show that MCSP requires depth-2 circuits (i.e., CNFs or DNFs) of size $2^{\Omega(N)}$, as in the case of PARITY?

What are other restricted models of computation against which we can show MCSP lower bounds using local PRGs? The recent "random walk PRG" by Chattopadhyay, Hatami, Hosseini, and Lovett [3] is also local and can be used to get MCSP lower bounds. However, as a general PRG that can be used to fool a variety of restricted models, it has sub-optimal usefulness (which is determined by its seed length) compared to the best-known lower bounds for most of those models.

References

1 Miklós Ajtai and Avi Wigderson. Deterministic Simulation of Probabilistic Constant Depth Circuits. *Advances in Computing Research*, 5:199–222, 1989. doi:10.1109/SFCS.1985.19.

2 Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from Random Strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006. doi:10.1137/050628994.

3 Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom Generators from Polarizing Random Walks. In *CCC*, pages 1:1–1:21, 2018. URL: https://eccc.weizmann.ac.il/report/2018/015/.

4 Mary Cryan and Peter Bro Miltersen. On Pseudorandom Generators in NC^0. In *Mathematical Foundations of Computer Science 2001, 26th International Symposium, MFCS 2001 Marianske Lazne, Czech Republic, August 27-31, 2001, Proceedings*, pages 272–284, 2001. doi:10.1007/3-540-44683-4_24.

5 Irit Dinur and Or Meir. Toward the KRW Composition Conjecture: Cubic Formula Lower Bounds via Communication Complexity. *Computational Complexity*, 27(3):375–462, 2018. doi:10.1007/s00037-017-0159-x.

6 Johan Håstad. Almost Optimal Lower Bounds for Small Depth Circuits. In *STOC*, 1986. doi:10.1145/12130.12132. doi:10.1145/12130.12132.

7 Johan Håstad. The Shrinkage Exponent of de Morgan Formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998. doi:10.1137/S0097539794261556.

8 Shuichi Hirahara. Non-Black-Box Worst-Case to Average-Case Reductions within NP. In *FOCS*, pages 247–258, 2018. URL: https://eccc.weizmann.ac.il/report/2018/138/.

9 Shuichi Hirahara and Rahul Santhanam. On the Average-Case Complexity of MCSP and Its Variants. In *CCC*, pages 7:1–7:20, 2017. doi:10.4230/LIPIcs.CCC.2017.7.

10 Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from Shrinkage. In *FOCS*, pages 111–119, 2012. URL: https://eccc.weizmann.ac.il/report/2012/057/.

11 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *STOC*, pages 73–79, 2000. URL: https://eccc.weizmann.ac.il/report/1999/045/.

12 E.I. Nechiporuk. On a Boolean function. *Doklady Akademii Nauk SSSR*, 169(4):765–766, 1966. English translation in Soviet Mathematics Doklady.

13 Noam Nisan and David Zuckerman. Randomness is Linear in Space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996. doi:10.1006/jcss.1996.0004.

14 Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:158, 2018. URL: https://eccc.weizmann.ac.il/report/2018/158/.

15 Igor Carboni Oliveira and Rahul Santhanam. Conspiracies Between Learning Algorithms, Circuit Lower Bounds, and Pseudorandomness. In *CCC*, pages 18:1–18:49, 2017. URL: https://eccc.weizmann.ac.il/report/2016/197/.

16 Igor Carboni Oliveira and Rahul Santhanam. Hardness Magnification for Natural Problems. In *FOCS*, pages 65–76, 2018. URL: https://eccc.weizmann.ac.il/report/2018/139/.

17 Claude E. Shannon. The Synthesis of Two-terminal Switching Circuits. *Bell Systems Technical Journal*, 28:59–98, 1949. doi:10.1002/j.1538-7305.1949.tb03624.x.

18 Avishay Tal. Shrinkage of De Morgan Formulae by Spectral Techniques. In *FOCS*, pages 551–560, 2014. URL: https://eccc.weizmann.ac.il/report/2014/048/.

19 Avishay Tal. Formula lower bounds via the quantum method. In *STOC*, pages 1256–1268, 2017. doi:10.1145/3055399.3055472.

20 Boris A. Trakhtenbrot. A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms. *IEEE Annals of the History of Computing*, 6(4):384–400, 1984. doi:10.1109/MAHC.1984.10036.

21 Luca Trevisan and Tongke Xue. A Derandomized Switching Lemma and an Improved Derandomization of AC^0. In *CCC*, pages 242–247, 2013. URL: https://eccc.weizmann.ac.il/report/2012/116/.

22 Ingo Wegener. *The complexity of Boolean functions*. Wiley-Teubner, 1987. URL: http://ls2-www.cs.uni-dortmund.de/monographs/bluebook/.