Electronic Colloquium on Computational Complexity, Report No. 170 (2015)

Weighted gate elimination: Boolean dispersers for quadratic varieties imply improved circuit lower bounds

Alexander Golovnev [*1] and Alexander S. Kulikov [†2]

[1]New York University
[2]St. Petersburg Department of Steklov Institute of Mathematics of the Russian Academy of Sciences

Abstract

In this paper we motivate the study of Boolean dispersers for quadratic varieties by showing that an explicit construction of such objects gives improved circuit lower bounds. An (n, k, s)-quadratic disperser is a function on n variables that is not constant on any subset of $\mathbb{F}_2^n$ of size at least s that can be defined as the set of common roots of at most k quadratic polynomials. We show that if a Boolean function f is a $\left(n, 1.83n, 2^{g(n)}\right)$-quadratic disperser for any function $g(n) = o(n)$ then the circuit size of f is at least $3.11n$. In order to prove this, we generalize the gate elimination method so that the induction works on the size of the variety rather than on the number of variables as in previously known proofs.

1 Introduction

1.1 Circuits and the gate elimination method

Denote by $B_{n,m}$ the set of all Boolean functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$, let $B_n = B_{n,1}$ and consider a function $f \in B_n$. A natural question studied in theoretical computer science is the following: what is the minimal number of binary Boolean operations needed to compute f? The corresponding computational model is Boolean circuits. A circuit is a directed acyclic graph with inputs $x_1, \ldots, x_n$, the intermediate vertices have in-degree 2 and are labeled with binary Boolean operations. The size of a circuit is its number of gates. Note that we do not pose any restrictions on the depth or out-degree. By $\mathcal{C}(f)$ we denote the minimum size of a circuit computing f.

Counting shows that the number of small size circuits is much smaller than the total number $|B_n| = 2^{2^n}$ of functions. Using this idea it was shown by Muller [16] that almost all functions from B_n require circuits of size $\Omega(2^n/n)$. This proof is however non-constructive: it does not give an explicit function with high circuit complexity. By saying explicit one usually means that the function is in P or NP. Finding an explicit function with high circuit complexity turned out to be an extremely difficult question. The currently strongest lower bound $3.011n$ was recently presented by Find et al. [10] improving a $3n - o(n)$ lower bound proved by Blum [3] more than 30 years ago.

[*]alexgolovnev@gmail.com
[†]kulikov@logic.pdmi.ras.ru

ISSN 1433-8092

Essentially, the only known technique for proving lower bounds for circuits with no restrictions on depth and out-degree is the gate elimination method. To illustrate it, we give a proof of a $2n - \Theta(1)$ lower bound given by Schnorr [17]. The $\mathrm{MOD}^n_{3,r} \in B_n$ function outputs 1 if and only if the sum (over integers) of n input bits is congruent to r modulo 3. We prove that $\mathrm{MOD}^n_{3,r}$ requires circuits of size at least $2n-6$ by induction on n. The base case $n \leq 3$ clearly holds. For the induction step consider an optimal circuit $\mathcal{C}$ computing $\mathrm{MOD}^n_{3,r}$ and its topologically minimal gate A (such a gate exists since for $n \geq 4$, $\mathrm{MOD}^n_{3,r}$ is not constant). Let x and y be input variables to A. The crucial observation is that either x or y must feed at least one other gate. Indeed if both x and y feed only A then the whole circuit depends on x and y only through A. This, in particular, means that by fixing x and y in four possible ways ($(x, y) = (0, 0), (0, 1), (1, 0), (1, 1)$) one gets at most two different subfunctions while there must be three different subfunctions under these assignments: $\mathrm{MOD}^{n-2}_{3,0}$, $\mathrm{MOD}^{n-2}_{3,1}$, and $\mathrm{MOD}^{n-2}_{3,2}$ (they are pairwise different for $n \geq 4$). Assume that it is x that feeds at least one other gate and call it B. We then replace x by 0. This eliminates at least two gates from the circuit (A and B): if one of the inputs to a gate computes a constant then this gate computes either a constant or a unary function on the other input and hence can be eliminated from the circuit. The resulting circuit computes the function $\mathrm{MOD}^{n-1}_{3,r}$ so the lower bound follows by induction. The best known lower bound for $\mathrm{MOD}^n_{3,r}$ is $2.5n - \Theta(1)$ by Stockmeyer [20], the best known upper bound is $3n + \Theta(1)$ by Demenkov et al. [6]. Knuth [12, solution to exercise 480] recently conjectured that the circuit size of $\mathrm{MOD}^n_{3,r}$ is equal to $3n - 5 - [(n + r) \bmod 3 = 0]$.

In the analysis above, we eliminated two gates by assigning $x \leftarrow 0$. If A computes, say, $xy = x \wedge y$ then we would have eliminated more than two gates since A becomes equal to 0 and hence all its successors are also eliminated. So, the bottleneck case is when both A and B compute parities of their inputs. In this case we cannot make A and B constant just by assigning a constant to x.

1.2 A $3n - o(n)$ lower bound for affine dispersers for sublinear dimension

A natural idea that allows to overcome the bottleneck from the previous subsection is to allow to substitute variables not only by constants but also by sums (over $\mathbb{F}_2$) of other variables. Using this idea one can prove a $3n - o(n)$ lower bound. The proof is due to Demenkov and Kulikov [7], the exposition here is due to Vadhan and Williams [21].

A function we are going to prove a lower bound for is called an affine disperser. Informally, an affine disperser is a function that cannot be made constant by sufficiently many linear substitutions. Formally, a function $f \in B_n$ is called an affine disperser for dimension d if it is not constant on any affine subspace of $\mathbb{F}_2^n$ of dimension at least d.

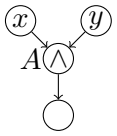
The notion of dispersers is a relaxation of the notion of extractors — functions that take input from some specific distribution and output a bit that is distributed statistically close to uniform. Unlike extractors, dispersers are only required to output a non-constant bit. To specify the class of input distributions, one defines a class of sources $\mathcal{F}$, where each $X \in \mathcal{F}$ is a distribution over $\mathbb{F}_2^n$. Since dispersers are only required to output a non-constant bit, we identify a distribution X with its support on $\mathbb{F}_2^n$. A function $f \in B_n$ is called a disperser for a class of sources $\mathcal{F}$, if $|f(X)| = 2$ for every $X \in \mathcal{F}$. Since it is impossible to extract even one non-constant bit from an arbitrary source (even if the source has almost full entropy), many special cases of sources are studied (see [19] for an excellent survey). The sources we are focused on in this paper are affine sources and their generalization — sources for polynomial varieties. Affine dispersers have drawn much interest lately. In particular, explicit constructions of affine dispersers for dimension $d = o(n)$ have been constructed [2, 22, 14, 18, 1, 15]. Dispersers for polynomial varieties over large fields

were studied by Dvir [8], and dispersers over $\mathbb{F}_2$ were studied by Cohen and Tal [5].

For a $3n-o(n)$ lower bound it is convenient to use xor-layered circuits. In an xor-layered circuit we allow linear sums of variables to be used as inputs to a circuit. Consider the following measure of an xor-layered circuit $\mathcal{C}$: $\mu(\mathcal{C}) = G(\mathcal{C})+I(\mathcal{C})$ where $G(\mathcal{C})$ is the number of non-input gates and $I(\mathcal{C})$ is the number of inputs of $\mathcal{C}$. Note that a xor-gate that depends on two inputs of an xor-layered circuit $\mathcal{C}$ may be replaced by an input without increasing $\mu(\mathcal{C})$.

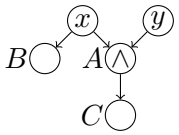
A $3n-4d$ lower bound for an affine disperser $f \in B_n$ for dimension d follows from the following fact: for any affine subspace $S \subseteq \mathbb{F}_2^n$ of dimension D and any xor-layered circuit $\mathcal{C}$ computing f on S, $\mu(\mathcal{C}) \geq 4(D-d-1)$. This can be shown by induction on D. The base case $D \leq d+1$ is clear. For the induction step, assume that $\mathcal{C}$ has the minimal value of μ. Let A be a top gate fed by linear sums x and y (such a gate must exist since f on S cannot compute a linear function as $D > d+1$). If A computes a sum of x and y then it can be replaced by an input (without increasing μ) so assume that A computes a product, i.e., $(x \oplus c_1)(y \oplus c_2) \oplus c$ where $c_1, c_2, c \in \mathbb{F}_2$ are constants. In the following we assign either $x = c_1$ or $y = c_2$. This gives us an affine subspace of $\mathbb{F}_2^n$ of dimension at least $D-1$ (if the dimension of the resulting subspace dropped to 0 this would mean that either x or y was constant on S contradicting to the fact that the considered circuit was optimal). To to proceed by induction we need to show that the substitution reduces μ by at least 4. For this, we consider two cases.

Case 1. Both x and y have out-degree 1.



We then assign $x = c_1$. This trivializes A to c, so all its successors are eliminated too. In total, we eliminate at least two gates (A and its successors) and at least two inputs (x and y). Hence μ is reduced by at least 4. (Note that A must have at least one successor as otherwise it would be an output gate, but this would mean that f was constant on an affine subspace of dimension at least d.)

Case 2. The out-degree of, say, x is at least 2.



Let B be another successor of x and let C be a successor of A. We assign $x = c_1$. This removes an input x and gates A, B, and C. If $B = C$ then C becomes a constant under the substitution (since both its inputs are constants) so its successors are also eliminated. Thus, in this case we eliminate at least one input and at least three gates implying that μ is reduced by at least 4.

Plugging in an affine disperser for sublinear dimension in this argument gives a $3n - o(n)$ lower bound. It is also interesting to note that the inequality $G(\mathcal{C}) + I(\mathcal{C}) \geq 4(n-d-1)$ is tight. To see this, note that the inner product function is an affine disperser for dimension $n/2+1$ (see, e.g., [4, Theorem A.1]) and has circuit size $n-1$.

1.3 Stronger lower bounds for dispersers for quadratic varieties

The two considered functions, MOD_3^n and an affine disperser, can be viewed as functions that are not constant on any sufficiently large set $S \subseteq \mathbb{F}_2^n$ that can be defined as the set of roots of k polynomials:

$$S = \{x \in \mathbb{F}_2^n \colon p_1(x) = p_2(x) = \cdots = p_k(x) = 0\}\,.$$

For MOD_3^n, $k \leq n-4$ and each p_i is just a variable or its negation while for affine dispersers, $k \leq n-d$ and p_i's are arbitrary linear polynomials. Note that the size of the set S can be easily determined from the number of polynomials in this case:

$$|S| = 2^{n-k}\,. \tag{1}$$

A natural extension is to allow polynomials to have degree at most 2. The corresponding set S is called a quadratic variety. Formally, a function $f \in B_n$ is called an (n,k,s)-quadratic disperser if it is not constant on any variety of size at least s defined by at most k polynomials of degree at most 2. The main result of this paper is the following.

Theorem 1. *Let $0 < \alpha \leq 1$ and $0 < \beta$ be constants satisfying*

$$2^{-\frac{2+\alpha}{\beta}} + 2^{-\frac{4+2\alpha}{\beta}} \leq 1, \tag{2}$$

$$2^{-\frac{2}{\beta}} + 2^{-\frac{5+2\alpha}{\beta}} \leq 1, \tag{3}$$

$$2^{-\frac{3+3\alpha}{\beta}} + 2^{-\frac{2+2\alpha}{\beta}} \leq 1, \tag{4}$$

$$2^{-\frac{3}{\beta}} + 2^{-\frac{4+\alpha}{\beta}} \leq 1, \tag{5}$$

and let $f \in B_{n,1}$ be an (n,k,s)-quadratic disperser. Then

$$\mathcal{C}(f) \geq \min\{\beta n - \beta \log_2 s - \beta, 2k\} - \alpha n\,.$$

For example, for an $(n, 1.83n, 2^{o(n)})$-quadratic disperser Theorem 1 with $\alpha = 0.535$ and $\beta = 3.6513$ implies a $3.1163n - o(n) > 3.116n$ lower bound. For an $(n, 1.78n, 2^{0.03n})$-quadratic disperser it implies a $3.006n$ lower bound.

Currently, explicit constructions of quadratic dispersers with such parameters are not known while showing their existence non-constructively is easy (see Lemma 1). Theorem 1 can be viewed as an additional motivation for their study.

Cohen and Tal [5] prove that any affine disperser (extractor) is also a disperser (extractor) for polynomial varieties with slightly weaker parameters. In particular, their result, combined with the affine disperser by Shaltiel [18], gives an explicit construction of an $\left(n, \Theta\left(\frac{n}{2^{\log^{0.9} n}}\right), 2^{o(n)}\right)$-quadratic disperser. Two explicit constructions of extractors for varieties over large fields are given by Dvir [8]. For a similar, although different, notion of polynomial sources, explicit constructions of dispersers (extractors) are given by Dvir, Gabizon, Wigderson [9] for large fields, and by Ben-Sasson and Gabizon [1] for constant-size fields.

1.4 Weighted gate elimination

We prove Theorem 1 by extending the gate elimination method. The proof goes by induction on the size of the current quadratic variety S. Note that for quadratic varieties the relation (1) no

longer holds: e.g., the set of roots of $n/2$ polynomials $x_1x_2 \oplus 1$, $x_3x_4 \oplus 1$, …, $x_{n-1}x_n \oplus 1$ contains just one point. For this reason, we proceed as follows. We choose a polynomial p of degree 2 and consider two subvarieties of S: $S_0 = \{x \in S\colon p(x) = 0\}$ and $S_1 = \{x \in S\colon p(x) = 1\}$. We then estimate how much the size of the circuit shrinks for each of these varieties and how much the size of the variety shrinks. Roughly, we show that in at least one of these cases the circuit shrinks a lot while the size of the variety does not shrink a lot. That is why we call this method weighted gate elimination.

2 Definitions

2.1 Circuits

A circuit is a directed acyclic graph with all nodes having in-degree 0 or 2. Nodes of in-degree 0 are labeled with input variables and are called inputs or input gates. Nodes of in-degree 2 are labeled with binary Boolean functions and are called gates or non-input gates. Some m gates are also marked as outputs. Then such a circuit computes a function from $B_{n,m}$ in a natural way.

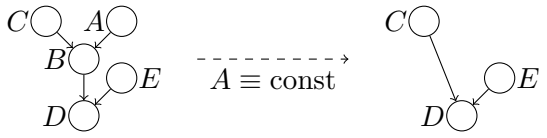
For a circuit $\mathcal{C}$, $G(\mathcal{C})$ is the number of non-input gates and is also called the size of the circuit $\mathcal{C}$. By $I(\mathcal{C})$ we denote the number of input gates. For a function $f \in B_{n,m}$, $\mathcal{C}(f)$ is the minimum size of a circuit with n inputs and m outputs that computes f. For a gate A, by $\mathrm{outdeg}(A)$ we denote the out-degree of A.

The 16 binary functions $b(x, y)$ from $B_{2,1}$ are usually classified as follows:

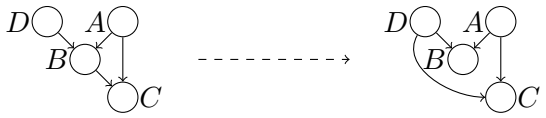
- 2 constant functions: 0, 1;
- 4 degenerate functions: x, $x \oplus 1$, y, $y \oplus 1$;
- 2 xor-type functions: $x \oplus y$, $x \oplus y \oplus 1$;
- 8 and-type functions: $(x \oplus a)(y \oplus b) \oplus c$ where $a, b, c \in \mathbb{F}_2$.

It is not difficult to see that gates computing constant and degenerate functions can be removed from a circuit. Hence an optimal circuit consists of gates computing xor-type functions and and-type functions. We call them xor-gates and and-gates, respectively.

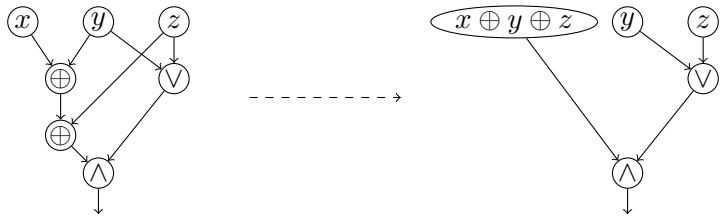
During the gate elimination process, we will make substitutions that make some gates constant. Assume that a gate A becomes constant (in this case, we also say that A is trivialized). Let B be a successor of A (that is, there is a directed edge from A to B), C be the other input of B, D be a successor of B, and E be the other input of D. Since A is now constant, B computes either a constant or a unary function on C so B can be eliminated. This may require also to change the binary function computed at D (that is, negating one of the inputs).



We will also use the following observation. Assume that C is fed by A and B while B is fed by A and D. Then C computes a binary function on A and D. This can be computed directly, without using B so one can rebuild the circuit as shown below. If B has out-degree 1 it can be eliminated from a circuit.



By an xor-layered circuit we mean a circuit whose inputs may be labeled not only by input variables but also by sums of variables. One can get an xor-layered circuit from a regular circuit by replacing xor-gates that depend on two inputs by an input.



2.2 Quadratic dispersers

Definition 1 (quadratic variety)**.** *A set $S \subseteq \mathbb{F}_2^n$ is called an* (n,k)-quadratic variety *if it can be defined as the set of common roots of $t \leq k$ polynomials of degree at most 2:*

$$S = \{x \in \mathbb{F}_2^n \colon p_1(x) = \cdots = p_t(x) = 0\}$$

where p_i is a polynomial of degree at most 2, for each $1 \leq i \leq t$.

Definition 2 (quadratic disperser)**.** *A Boolean function $f \in B_n$ is called an* (n,k,s)-quadratic disperser *if f is non-constant on any (n,k)-quadratic variety $S \subseteq \mathbb{F}_2^n$ of size at least s.*

The following lemma shows that almost all functions from B_n are $(n, 2^{o(n)}, 2^{o(n)})$-quadratic dispersers.

Lemma 1. *Let $\omega(1) \leq s \leq 2^{o(n)}, k = o\left(\frac{s}{n^2}\right)$. Let $D_n \in B_{n,1}$ be the set of (n,k,s)-quadratic dispersers. Then $\frac{|D_n|}{|B_n|} \to 1$ when $n \to \infty$.*

Proof. There are $q = \frac{n(n+1)}{2} + 1 = \Theta(n^2)$ monomials of degree at most 2 in $\mathbb{F}_2^n$. Therefore, there are 2^q polynomials of degree at most 2, and at most 2^{qk} (n,k)-quadratic varieties. Each function that is not an (n,k,s)-quadratic disperser can be specified by

1. an (n,k)-quadratic variety, where it takes a constant value,
2. one of two possible constant values that it takes on that variety,
3. values at the remaining at most $2^n - s$ points.

Thus, the number of functions that are not (n,k,s)-quadratic dispersers is bounded from above by $2^{qk} \cdot 2 \cdot 2^{2^n - s} = 2^{2^n} 2^{qk+1-s} = 2^{2^n} 2^{-\Theta(s)} = o\left(|B_n|\right)$. □

3 Lower bound

We will use the following technical lemma.

Lemma 2. *Let $0 < \alpha \le 1$ and $0 < \beta$ be constants satisfying inequalities (5), (2):*

$$\begin{aligned} 2^{-\frac{3}{\beta}} + 2^{-\frac{4+\alpha}{\beta}} &\le 1, \\ 2^{-\frac{2+\alpha}{\beta}} + 2^{-\frac{4+2\alpha}{\beta}} &\le 1. \end{aligned}$$

Then

$$2^{-\frac{4}{\beta}} + 2^{-\frac{4}{\beta}} \le 1, \tag{6}$$

$$2^{-\frac{3+\alpha}{\beta}} + 2^{-\frac{3+2\alpha}{\beta}} \le 1. \tag{7}$$

Proof. Since $2 \le x + \frac{1}{x}$ for positive x,

$$2^{-\frac{4}{\beta}} + 2^{-\frac{4}{\beta}} \le 2^{-\frac{4}{\beta}}(2^{\frac{1}{\beta}} + 2^{-\frac{1}{\beta}}) = 2^{-\frac{3}{\beta}} + 2^{-\frac{5}{\beta}} \le 2^{-\frac{3}{\beta}} + 2^{-\frac{4+\alpha}{\beta}} \le 1\,.$$

In order to prove the inequality (7), we use Heinz's inequality [11]:

$$\frac{x^{1-t}y^t + x^t y^{1-t}}{2} \le \frac{x+y}{2} \text{ for } x, y > 0, 0 \le t \le 1.$$

Let us take $x = 2^{-\frac{2+\alpha}{\beta}}, y = 2^{-\frac{4+2\alpha}{\beta}}, t = \frac{1}{2+\alpha}$:

$$2^{-\frac{3+\alpha}{\beta}} + 2^{-\frac{3+2\alpha}{\beta}} = x^{1-t}y^t + x^t y^{1-t} \le x + y = 2^{-\frac{2+\alpha}{\beta}} + 2^{-\frac{4+2\alpha}{\beta}} \le 1.$$

□

In the following lemma, we use the following circuit complexity measure: $\mu(\mathcal{C}) = G(\mathcal{C}) + \alpha \cdot I(\mathcal{C})$ where $0 < \alpha \le 1$ is a constant to be determined later. Theorem 1 follows from the lemma with $S = \mathbb{F}_2^n$ which is an $(n, 0)$-quadratic variety.

Lemma 3. *Let $f \in B_n$ be an (n, k, s)-quadratic disperser, $S \subseteq \mathbb{F}_2^n$ be an (n, t)-quadratic variety, $0 < \alpha \le 1, 0 < \beta$ be constants satisfying inequalities (2), (3), (4), (5), $\mathcal{C}$ be an xor-layered circuit that computes f on S. Then*

$$\mu(\mathcal{C}) \ge \min\left\{\beta(\log_2 |S| - \log_2 s - 1), 2(k-t)\right\}.$$

Proof. The proof goes by induction on $|S|$. The base case $|S| \le 2s$ is trivially true. For the induction step, assume that $|S| > 2s$.

To prove the induction step we proceed as follows. If $t \le k$ then the right-hand side is non-positive, so assume that $t > k$. Assume that $\mathcal{C}$ is optimal with respect to μ (that is, $\mathcal{C}$ has the minimal value of μ among all circuits computing f on S). We find a gate G in $\mathcal{C}$ that computes a function g of degree at most 2 and consider two $(n, t+1)$-quadratic varieties of S: $S_0 = \{x \in S \colon g(x) = 0\}$ and $S_0 = \{x \in S \colon g(x) = 1\}$. Let $|S_0| = p_0|S|$ and $|S_1| = p_1|S|$ where $0 < p_0, p_1 < 1$ and $p_0 + p_1 = 1$ (note that $p_i = 0$ or $p_i = 1$ would mean that G computes a constant on S contradicting to the fact that C is optimal). By eliminating from the circuit C all the gates that

are either constant or depend on just one of its inputs on S_i, one gets a circuit $\mathcal{C}_i$ that computes f on S_i. Assume that $\mu(\mathcal{C}) - \mu(\mathcal{C}_i) \geq \Delta_i$. Then, by the induction hypothesis,

$$\begin{aligned}\mu(C) &\geq \mu(C_i) + \Delta_i \geq \min\{\beta(\log_2|S_i| - \log_2 s - 1), 2(k-(t+1))\} + \Delta_i \\ &= \min\left\{\beta\left(\log_2|S| - \log_2 s - 1\right) + \left(\Delta_i - \beta\log_2\frac{1}{p_i}\right), 2(k-t) + (\Delta_i - 2)\right\}.\end{aligned}$$

Hence, if $\Delta_i \geq \beta\log_2 1/p_i$ and $\Delta_i \geq 2$ for either $i=0$ or $i=1$ then the required inequality follows by the induction hypothesis. The inequality $\Delta_i \geq \beta\log_2 1/p_i$ is true whenever $p_i \geq 2^{-\frac{\Delta_i}{\beta}}$. Since we want this inequality to hold for at least one of $i=0$ and $i=1$ and since $p_0 + p_1 = 1$ we conclude that for the induction step to go through it suffices to have

$$2^{-\frac{\Delta_0}{\beta}} + 2^{-\frac{\Delta_1}{\beta}} \leq 1 \text{ and } \Delta_0, \Delta_1 \geq 2\,. \tag{8}$$

By going through a few cases we show that we can always find a gate G such that the corresponding Δ_0 and Δ_1 satisfy the inequalities (8). For this, we use the inequalities (2)–(7).

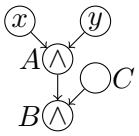
We start by showing that the circuit $\mathcal{C}$ must be non-empty. Indeed, if $\mathcal{C}$ is empty then it computes a linear function l. Hence f is constant on both $S_0 = \{x \in S\colon l(x) = 0\}$ and $S_1 = \{x \in S\colon l(x) = 1\}$. However $\max\{|S_0|, |S_1|\} \geq |S|/2 > s$ which contradicts to the fact that f is an (n,k,s)-quadratic disperser.

Let A be an and-gate with the maximal number of and-gates on a way to the output of $\mathcal{C}$. That is, for each and-gate we consider all directed paths from this gate to the output gate and select a path with the maximal number of and-gates on it; then we choose an and-gate for which this number is maximal over all and-gates. Since $\mathcal{C}$ is an xor-layered circuit, we may assume that A is a top-gate, that is, it is fed by inputs. Denote by x and y the input-gates that feed A.

Case 1. $\mathrm{outdeg}(x) = \mathrm{outdeg}(y) = 1$.

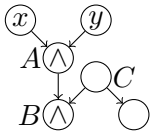
Case 1.1. $\mathrm{outdeg}(A) = 1$ and A feeds an and-gate B.
Let C be the other input of B (it might be an input as well as non-input gate).

Case 1.1.1. $\mathrm{outdeg}(C) = 1$.



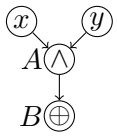
We make A constant. Then the gate B is eliminated. Moreover, either $A = 0$ or $A = 1$ trivializes the gate B so all its successors and the gate C are also eliminated (since C is only used to compute B, but B now computes a constant). In both cases x and y are not needed anymore (as the only gate A that was fed by both these inputs is now constant). So, we get $\{\Delta_0, \Delta_1\} = \{2 + 2\alpha, 3 + 3\alpha\}$. The required inequalities (8) follows from (4).

Case 1.1.2. $\mathrm{outdeg}(C) \geq 2$.



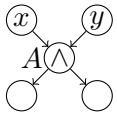
Because of the choice of A, C computes a function of degree at most 2. We make C constant. In both cases we eliminate two successors of C and C itself. This reduces the measure by at least $2+\alpha$. In one of the cases B is trivialized which causes the removal of the successors of B, the gate A, and inputs x and y. Hence we get $\{\Delta_0, \Delta_1\} = \{2+\alpha, 4+3\alpha\}$ in this case. These Δ_0, Δ_1 satisfy the inequalities (8) because of (2).

Case 1.2. outdeg$(A) = 1$ and A feeds an xor-gate B.



Since A was chosen as an and-gate with the maximal number of and-gates to the output, the other input of B computes a function of degree at most 2. Hence B itself computes a function of degree at most 2. We make B constant. This eliminates B and its successors. The gate A and its inputs x and y are also not needed. Hence $\Delta_0 = \Delta_1 = 3+2\alpha$. The inequalities (8) are satisfied due to (7).

Case 1.3. outdeg$(A) \geq 2$.



Just by making the gate A constant we get $\Delta_0 = \Delta_1 = 3+2\alpha$ since A and all its successors (at least two gates) are eliminated. Similarly to the previous case, the inequality (7) imply that (8) holds.

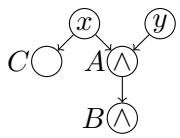
Case 2. Out-degree of either x or y is at least 2. Say, outdeg$(x) \geq 2$.

Case 2.1. outdeg$(A) = 1$ and A feeds an and-gate B.

We make A constant. Assume that A computes $(x \oplus c_1)(y \oplus c_2) \oplus c$. Then A can only be equal to $c \oplus 1$ if $x = c_1 \oplus 1$ and $y = c_2 \oplus 1$. That is, when A is equal to $c \oplus 1$ not only its successor is eliminated but also all successors of x and y. In both cases the gate B is eliminated, but in one of them it is trivialized and so all its successors are also eliminated.

Denote by C another gate fed by x. Note that $B \neq C$ (otherwise the circuit would not be optimal).

Case 2.1.1. outdeg$(y) = 1$.



Case 2.1.1.1. B is trivialized when $A = c$.

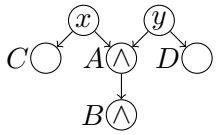
If $A = c$ we eliminate A, B, the successors of B, and y. If $A = c \oplus 1$ we eliminate A, B, C, x, and y. Hence $\{\Delta_0, \Delta_1\} = \{3+\alpha, 3+2\alpha\}$. The inequality (7) guarantees that (8) holds.

Case 2.1.1.2. B is trivialized when $A = c \oplus 1$.

If $A = c$ we eliminate A, B, and y. If $A = c \oplus 1$ we eliminate A, B, C, the successors of B, x, and y (if C happens to be the only successor of B then it becomes constant and all its successors are eliminated). Hence $\{\Delta_0, \Delta_1\} = \{2 + \alpha, 4 + 2\alpha\}$. The inequalities (8) are satisfied because of (2).

Case 2.1.2. $\text{outdeg}(y) \geq 2$.

Denote by D another successor of y. Note that D might be equal to C, but $D \neq B$.



Case 2.1.2.1. B is trivialized when $A = c$.

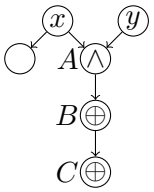
If $A = c$ we eliminate A, B, and the successors of B. If $A = c \oplus 1$ we eliminate A, B, C, D, x, and y. If $C = D$ then this gate becomes constant so all its successors are also eliminated. Hence $\{\Delta_0, \Delta_1\} = \{3, 4 + 2\alpha\}$. The inequalities (8) are satisfied because (5).

Case 2.1.2.2. B is trivialized when $A = c \oplus 1$.

If $A = c$ we eliminate A and B. If $A = c \oplus 1$ we eliminate A, B, C, D, the successors of B, x, and y. In this case we need to take additional care to show that we eliminate five gates even if some of the mentioned five gates coincide. If $C \neq D$ and, say, C is a successor of B then C becomes constant so all its successors are eliminated too. If $C = D$ then C becomes constant so all its successors are eliminated. Hence $\{\Delta_0, \Delta_1\} = \{2, 5 + 2\alpha\}$. The inequality (3) ensures (8).

Case 2.2. $\text{outdeg}(A) = 1$ and A feeds an xor-gate B.

Case 2.2.1. $\text{outdeg}(B) = 1$ and B feeds an xor-gate C.

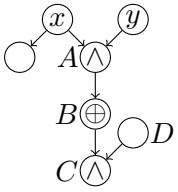


Because of the choice of A, we know that the gate C computes a quadratic function. We make C constant. In both cases we eliminate A, B, C, and the successors of C. Hence $\Delta_0 = \Delta_1 = 4$. The inequalities (8) are satisfied because of (6).

Case 2.2.2. $\text{outdeg}(B) = 1$ and B feeds an and-gate C.

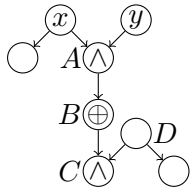
Let D be the other input of C. Note that if $D = A$ then the circuit is not optimal (C depends on A and the other input of B so one can compute C directly without using B).

Case 2.2.2.1. $\text{outdeg}(D) = 1$.



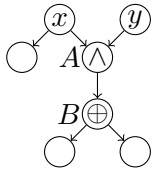
We make B constant. In both cases we eliminate A, B, and C. Moreover, when B is the constant trivializing C we eliminate also D and the successors of C. The gate D contributes (to the complexity decrease) $\alpha \leq 1$ if it is an input gate and 1 if it is not an input. Hence we have $\{\Delta_0, \Delta_1\} = \{3, 4+\alpha\}$. The inequality (5) guarantees that (8) is satisfied.

Case 2.2.2.2. outdeg$(D) \geq 2$.



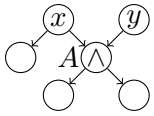
We make D constant (we are allowed to do so because it computes a function of degree at most 2). In both cases we eliminate D and its successors and reduce the measure by at least $2+\alpha$ (as D might be an input). In the case when C becomes constant we eliminate also the successors of C as well as A and B. Thus, $\{\Delta_0, \Delta_1\} = \{2+\alpha, 5+\alpha\}$ (to ensure that all the five gates eliminated in the second case are different one notes that if D feeds B or a successor of C then the circuit is not optimal). The inequalities (8) are satisfied because (2) and $\alpha \leq 1$.

Case 2.2.3. outdeg$(B) \geq 2$.



The gate B computes a function of degree at most 2. By making it constant we eliminate B, its successors, and A, so $\Delta_0 = \Delta_1 = 4$. The inequalities (8) are satisfied because of (6).

Case 2.3. outdeg$(A) \geq 2$.



We make A constant. In both cases A and its successors are eliminated. When x and y become constant too (recall that if A computes $(x \oplus c_1)(y \oplus c_2) \oplus c$ then $A = c \oplus 1$ implies that $x = c_1 \oplus 1$ and $y = c_2 \oplus 1$) at least one other successor of x is also eliminated. Thus, $\{\Delta_0, \Delta_1\} = \{3, 4+2\alpha\}$. The inequality (5) implies that (8) is satisfied.

□

4 Lower bounds for multi-output functions

Note that $3.011n + o(n)$ is also the currently strongest lower bound even for functions from $B_{n,o(n)}$ (that is, functions with $o(n)$ outputs). Strongest known lower bounds for multi-output functions

from $B_{n,m}$ follow from the following lemma by Lamagne and Savage [13]. It can be read as follows: if instead of one function one needs to compute m functions then at least $m-1$ additional gates are needed.

Lemma 4. *Let $f=(f_1,\dots,f_m)\in B_{n,m}$ such that $f_i\not\equiv f_j$ and $f_i\not\equiv f_j\oplus 1$ for all $1\le i\ne j\le m$. Then*

$$\mathcal{C}(f)\ge \min_{1\le i\le m}\mathcal{C}(f_i)+(m-1)\,.$$

Proof sketch. Fix a topological ordering of the gates of a circuit and consider the first $\min_{1\le i\le m}\mathcal{C}(f_i)-1$ gates in this ordering. None of them can compute any of m output functions. Since all the output functions are different, at least m additional gates are needed to compute all of them. □

Thus, it would be interesting to give an explicit construction of a quadratic disperser with good parameters (implying a stronger than $3.011n$ lower bound on circuit complexity) with $o(n)$ outputs. Note that almost all known explicit constructions of dispersers are actually multi-output functions $f\in B_{n,m}$ that output $(1-\varepsilon)\cdot 2^m$ different values on each source. For our purposes, it is enough for a disperser to have at least 2 different values (that is, to be non-constant). Such a disperser is a weaker object than even a single-output disperser, so it might be easier to construct it. For example such a relaxation helps to construct a disperser for polynomial sources (a similar, but still different from polynomial varieties, notion of sources) over smaller fields [1].

5 Open problems and further directions

One can slightly improve the lower bound for quadratics dispersers by a more involved case analysis. Dispersers for varieties of degree 3 allow to get even stronger lower bounds. At the same time we do not see how the presented techniques might lead to, say, a lower bound of $4n$.

The most natural question left open by this study is: to find an explicit construction of an $(n,1.78n,2^{0.03n})$-quadratic disperser either with $o(n)$ outputs or with one output. Note that such a construction would automatically imply a new circuit lower bound. It would also be interesting to find explicit constructions of dispersers for polynomial varieties of higher degrees, as well as their applications to circuit lower bounds.

Acknowledgements

The research leading to these results received funding from National Science Foundation under Grant 1319051 and the Government of the Russian Federation under Grant 14.Z50.31.0030.

References

[1] Eli Ben-Sasson and Ariel Gabizon. Extractors for polynomials sources over constant-size fields of small characteristic. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 399–410. Springer, 2012.

[2] Eli Ben-Sasson and Swastik Kopparty. Affine dispersers from subspace polynomials. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, volume 679, pages 65–74. ACM Press, 2009.

[3] Norbert Blum. A Boolean function requiring $3n$ network size. *Theoretical Computer Science*, 28:337–345, 1984.

[4] Gil Cohen and Igor Shinkar. The complexity of DNF of parities. Technical Report 99, Electronic Colloquium on Computational Complexity, 2014.

[5] Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. *arXiv preprint arXiv:1404.0654*, 2014.

[6] Evgeny Demenkov, Arist Kojevnikov, Alexander S. Kulikov, and Grigory Yaroslavtsev. New upper bounds on the Boolean circuit complexity of symmetric functions. *Information Processing Letters*, 110(7):264–267, 2010.

[7] Evgeny Demenkov and Alexander S. Kulikov. An elementary proof of a $3n - o(n)$ lower bound on the circuit complexity of affine dispersers. In *Proceedings of 36th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 6907 of *Lecture Notes in Computer Science*, pages 256–265. Springer, 2011.

[8] Zeev Dvir. Extractors for varieties. *Computational complexity*, 21(4):515–572, 2012.

[9] Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18(1):1–58, 2009.

[10] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-$3n$ lower bound for the circuit complexity of an explicit function. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:166, 2015.

[11] Erhard Heinz. Beiträge zur störungstheorie der spektralzerleung. *Mathematische Annalen*, 123(1):415–438, 1951.

[12] Donald E. Knuth. *The Art of Computer Programming*, volume 4, pre-fascicle 6a. Addison–Wesley, 2015. Section 7.2.2.2. Satisfiability. Draft available at http://www-cs-faculty.stanford.edu/~uno/fasc6a.ps.gz.

[13] Edward A. Lamagna and John E. Savage. On the logical complexity of symmetric switching functions in monotone and complete bases. Technical report, Brown University, 1973.

[14] Xin Li. A new approach to affine extractors and dispersers. In *Proceedings of 26th Annual Conference on Computational Complexity (CCC)*, pages 137–147. IEEE, 2011.

[15] Xin Li. Extractors for affine sources with polylogarithmic entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:121, 2015.

[16] David E. Muller. Complexity in electronic switching circuits. *IRE Transactions on Electronic Computers*, EC-5:15–19, 1956.

[17] Claus-Peter Schnorr. Zwei lineare untere Schranken für die Komplexität Boolescher Funktionen. *Computing*, 13:155–171, 1974.

[18] Ronen Shaltiel. Dispersers for affine sources with sub-polynomial entropy. In *Proceedings of 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–256. IEEE, 2011.

[19] Ronen Shaltiel. An introduction to randomness extractors. In *Proceedings of 38th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 6756 of *Lecture Notes in Computer Science*, pages 21–41. Springer, 2011.

[20] Larry J. Stockmeyer. On the combinational complexity of certain symmetric Boolean functions. *Mathematical Systems Theory*, 10:323–336, 1977.

[21] Salil Vadhan and Ryan Williams. Personal communication, 2013.

[22] Amir Yehudayoff. Affine extractors over prime fields. *Combinatorica*, 31(2):245–256, 2011.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de