Electronic Colloquium on Computational Complexity, Report No. 184 (2014)

Correlation Bounds and #SAT Algorithms for Small Linear-Size Circuits

Ruiwen Chen[*] Valentine Kabanets[†]

December 28, 2014

Abstract

We revisit the gate elimination method, generalize it to prove correlation bounds of boolean circuits with Parity, and also derive deterministic #SAT algorithms for small linear-size circuits. In particular, we prove that, for boolean circuits of size $3n - n^{0.51}$, the correlation with Parity is at most $2^{-n^{\Omega(1)}}$, and there is a #SAT algorithm running in time $2^{n-n^{\Omega(1)}}$; for circuit size $2.99n$, the correlation with Parity is at most $2^{-\Omega(n)}$, and there is a #SAT algorithm running in time $2^{n-\Omega(n)}$. Similar correlation bounds and algorithms are also proved for circuits of size almost $2.5n$ over the full binary basis B_2.

Keywords: boolean circuit, random restriction, correlation bound, satisfiability algorithm.

1 Introduction

Connections between circuit lower bounds and efficient algorithms have been explicitly exploited in several recent breakthroughs. In particular, the "random restriction" technique, which was used to prove circuit lower bounds, was extended to get both satisfiability algorithms and average-case lower bounds for boolean formulas [San10, KR13, KRT13, CKK$^+$14] and AC^0 circuits [IMP12, BIS12].

For de Morgan formulas, Santhanam [San10] gave a #SAT algorithm running in time $2^{n-\Omega(n)}$ for formulas of linear size; the algorithm is based on a generalization of the "shrinkage under random restrictions" property, which was used to prove formula lower bounds [Sub61, Hås98]. Santhanam [San10] observed that, one can define a random process of restrictions such that the formula size shrinks with high probability. This *concentrated shrinkage* implies not only #SAT algorithms but also correlation bounds. As shown in [San10], a linear-size de Morgan formula has correlation at most $2^{-\Omega(n)}$ with Parity; the correlation of two n-input functions f and g is $|\mathbf{Pr}[f(x) = g(x)] - \mathbf{Pr}[f(x) \neq g(x)]|$, where x is chosen uniformly at random from $\{0,1\}^n$. Santhanam's algorithm was extended to $2^{n-n^{\Omega(1)}}$-time #SAT algorithms for de Morgan formulas of size $n^{2.49}$ in [CKK$^+$14] and size $n^{2.63}$ in [CKS14]. For formulas over the full binary basis B_2, Seto and Tamaki [ST12] extended [San10] to give a $2^{n-\Omega(n)}$-time #SAT algorithm for B_2-formulas of linear size, and also showed that such formulas cannot approximately compute affine extractors.

On the other hand, Komargodski, Raz, and Tal [KR13, KRT13] also used the concentrated shrinkage property to generalize the worst-case formula lower bounds to the average case. They

[*]School of Informatics, University of Edinburgh, Edinburgh, UK; rchen2@inf.ed.ac.uk
[†]School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada; kabanets@cs.sfu.ca

ISSN 1433-8092

Table 1: Worst-case and average-case lower bounds for computing Parity

	Worst-Case Lower Bounds	Average-Case Upper / Lower Bounds	
AC^0	$s = \exp(n^{\theta(\frac{1}{d-1})})$ [Yao85, Hås86]	$\epsilon = 2^{-\Omega(n/(\log s)^{d-1})}$ [Hås12]	
De Morgan formulas	$s = n^{2-\theta(1)}$ [Sub61]	$\epsilon \geqslant 2^{-\Omega(n^2/s)}$	$\epsilon \leqslant 2^{-\Omega(n/\sqrt{s})}$ [BBC+01, Rei11] $\epsilon \leqslant 2^{-\Omega(n/c^2)}$ for $s = cn$ [San10]
U_2-circuits	$s = 3n - \theta(1)$ [Sch74]	$\epsilon \geqslant 2^{-\Omega(3n-s)}$	$\epsilon \leqslant 2^{-\Omega((3n-s)^2/n)}$ [This work]

gave an explicit function (computable in polynomial time) such that de Morgan formulas of size $n^{2.99}$ can compute correctly on at most $1/2 + 2^{-n^{\Omega(1)}}$ fraction of inputs. Combining the techniques in [KRT13, CKK+14], one can get a randomized $2^{n-n^{\Omega(1)}}$-time #SAT algorithm for de Morgan formulas of size $n^{2.99}$.

1.1 Our results and techniques

In this work, we get correlation bounds and #SAT algorithms for general boolean circuits. We consider circuits over the full binary basis B_2 and circuits over the basis $U_2 = B_2 \setminus \{\oplus, \equiv\}$.

We prove that, for U_2-circuits of size $3n - n^\epsilon$ for $\epsilon > 0.5$, the correlation with Parity is at most $2^{-n^{\Omega(1)}}$, and there is a #SAT algorithm running in time $2^{n-n^{\Omega(1)}}$; for U_2-circuits of size $3n - \epsilon n$ for $\epsilon > 0$, the correlation is at most $2^{-\Omega(n)}$, and there is a #SAT algorithm running in time $2^{n-\Omega(n)}$. For B_2-circuits, we give a similar #SAT algorithm for circuits of size almost $2.5n$, and show the average-case hardness of computing affine extractors using such circuits.

Our correlation bounds of U_2-circuits with Parity are almost optimal, up to constant factors in the exponents. In fact, one can construct a U_2-circuit of size $3n - l$ which computes Parity on at least $1/2 + 2^{-\Omega(l)}$ fraction of inputs. Table 1 summarizes the known worst-case and average-case lower bounds against Parity for several restricted circuit models. Note that, for the average-case bounds, we express the correlation ϵ as a function of the circuit size s.

However, there is still a gap between our average-case lower bounds and the worst-case lower bounds. The best known worst-case explicit lower bound is $5n - o(n)$ for U_2-circuits [LR01, IM02], and $3n - o(n)$ for B_2-circuits [Blu84].

For #SAT algorithms, there is a known algorithm for B_2-circuits by Nurk [Nur09] which runs in time $O(2^{0.4058s})$ for circuits of size s. The running time of our algorithm for B_2-circuits is almost the same as Nurk's [Nur09]. We are not aware of any #SAT algorithm for U_2-circuits.

Our techniques. We extend the gate elimination method which was previously used to prove worst-case circuit lower bounds [Sch74, Blu84, Zwi91, LR01, IM02, DK11]. We define a random process of restrictions such that the circuit size shrinks with high probability. This is similar to the concentrated shrinkage approach for boolean formulas [San10, ST12, KR13, KRT13, CKK+14]. We analyze this random process using the concentration bound given by a variant of Azuma's inequality as in [CKK+14]. This analysis is then used to get both correlation bounds and #SAT algorithms. The same approach works for both U_2-circuits and B_2-circuits, although we need different rules on defining restrictions.

As a byproduct of our algorithms, we show that small linear-size circuits have decision trees of non-trivial size. In particular, U_2-circuits of size s have equivalent decision trees of size $2^{n-\Omega((3n-s)^2/n)}$, and B_2-circuits of size s have parity decision trees of size $2^{n-\Omega((2.5n-s)^2/n)}$. Our correlation bounds follow directly from such non-trivial decision-tree representations.

Related work. For U_2-circuits, the best known worst-case lower bound is $5n - o(n)$ by Iwama and Morizumi [IM02], improving upon a $4.5n - o(n)$ lower bound by Lachish and Raz [LR01], a $4n - c$ lower bound against symmetric functions by Zwick [Zwi91], and a $3n - c$ lower bound against Parity by Schnorr [Sch74]. For B_2-circuits, the best known worst-case lower bound is $3n - o(n)$ by Blum [Blu84]; Demenkov and Kulikov [DK11] gives an alternative proof of this lower bound against affine dispersers. Nurk [Nur09] gave a satisfiability algorithm running in time $O(2^{0.4058s})$ for B_2-circuits of size s. Nurk's algorithm [Nur09] is also based on gate elimination and the running time is similar to ours, although we use a slightly different case analysis for gate elimination. We are not aware of any previous average-case lower bounds (correlation bounds) for general circuits.

2 Preliminaries

2.1 Circuits

Let B be a binary basis, i.e., a set of boolean functions on two variables. A *B-circuit* on n input variables is a directed acyclic graph with (1) nodes of in-degree 0 labeled by variables or constants, which we call *inputs*, and (2) nodes of in-degree 2 labeled by functions from B, which we call *gates*. There is a single node of out-degree 0, designated as the *output*. Without loss of generality, we assume, for each variable x_i, there is at most one input labeled by x_i. A circuit on n variables computes a boolean function $f\colon \{0,1\}^n \to \{0,1\}$. For two nodes u and v, we will write $u \to v$ if u feeds into v.

We consider two binary bases: the full basis B_2, which contains all boolean functions on two variables, and the basis $U_2 = B_2 \setminus \{\oplus, \equiv\}$. Specifically, the basis B_2 contains the following 16 functions $f(x, y)$:

- six degenerate functions: $0,\ 1,\ x,\ \neg x,\ y,\ \neg y$;
- eight $\wedge$-type functions: $x \wedge y,\ x \vee y$, and the variations by negating one or both inputs;
- two $\oplus$-type functions: $x \oplus y,\ x \equiv y$.

The *size* of a circuit C, denoted by $s(C)$, is the number of gates in C. The *circuit size* of a function $f\colon \{0,1\}^n \to \{0,1\}$ is the minimal size of a boolean circuit computing f. For convenience, we define $\mu(C) = s(C) + N(C)$, where $N(C)$ is the number of inputs that C depends on. We let $\mu(C) = 0$ if C is constant, and $\mu(C) = 1$ if C is a literal.

A *restriction* ρ is a mapping from the input variables to $\{0, 1, *\}$. For a circuit C, the restricted circuit $C|_\rho$ is obtained by fixing $x_i = b$ for all x_i such that $\rho(x_i) = b \in \{0,1\}$.

It is convenient to work with circuits without redundant nodes or wires. We will call a non-constant circuit (over U_2 or B_2) *simplified* if it does not have the following:

1. nodes labeled by constants,
2. gates labeled by degenerate functions,
3. non-output gates with out-degree 0, or

4. any input x and two gates u, v with three wires $x \to u$, $x \to v$, $u \to v$.

Lemma 2.1. *For any circuit C, there is a polynomial-time algorithm transforming C into an equivalent simplified circuit C' such that $s(C') \leqslant s(C)$ and $\mu(C') \leqslant \mu(C)$.*

Proof Sketch. Cases (1)-(3) are trivial. For case (4), suppose w is the other node feeding into u. If C is over B_2, then v computes a binary function of x and w; if C is over U_2, then v computes an $\wedge$-type function of x and w (because a $\oplus$-type function requires at least 3 gates). In either case, we can connect w directly to v, remove the wire $u \to v$, and change the gate label of v. By checking through each input and gate, the transformation can be done in polynomial time. $\square$

2.2 Correlation

Definition 2.2. Let f and g be two boolean functions on n input variables. The *correlation* of f and g is defined as

$$\mathrm{Corr}(f, g) = |\mathbf{Pr}[f(x) = g(x)] - \mathbf{Pr}[f(x) \neq g(x)]| = |2\mathbf{Pr}[f(x) = g(x)] - 1|\,,$$

where x is chosen uniformly at random from $\{0,1\}^n$.

The *correlation* of f with a circuit class $\mathcal{C}$ is the maximum of $\mathrm{Corr}(f, C)$ for any $C \in \mathcal{C}$. Note that, a circuit C has correlation c with f if and only if C computes f or its negation correctly on a fraction $(1+c)/2$ of all inputs. The correlation bound is also referred to as the *average-case lower bound* in the literature.

2.3 Decision Tree

A *decision tree* is a tree where (1) each internal node is labeled by a variable x, and has two outgoing edges labeled by $x = 0$ and $x = 1$, and (2) each leaf is labeled by a constant 0 or 1. A decision tree computes a boolean function by tracking the paths from the root to leaves. The *size* of a decision tree is the number of leaves of the tree.

A *parity decision tree* extends a decision tree such that each internal node is labeled by the parity of a subset of variables (including one single variable as a special case). We insist that, for each path from the root to a leaf, the parities appearing in the internal nodes are linearly independent.

2.4 Concentration bounds

Theorem 2.3 (Chernoff bounds)**.** *[AB09] Let $\{X_i\}_{i=1}^n$ be mutually independent random variables over $\{0,1\}$, and let $\mu = \sum_{i=1}^n \mathbf{E}[X_i]$. Then, for every $c > 0$,*

$$\mathbf{Pr}\left[\left|\sum_{i=1}^n X_i - \mu\right| \geqslant c\mu\right] \leqslant 2 \cdot e^{-\min\{c^2/4,\ c/2\}\mu}.$$

A sequence of random variables $X_0, X_1, \ldots, X_n$ is called a *supermartingale* with respect to a sequence of random variables $R_1, \ldots, R_n$ if $\mathbf{E}[X_i \mid R_{i-1}, \ldots, R_1] \leqslant X_{i-1}$, for $1 \leqslant i \leqslant n$. The following is a variant of Azuma's inequality which holds for supermartingales with one-side bounded differences.

Lemma 2.4. *[CKK+14] Let $\{X_i\}_{i=0}^n$ be a supermartingale with respect to $\{R_i\}_{i=1}^n$. Let $Y_i = X_i - X_{i-1}$. If, for every $1 \leqslant i \leqslant n$, the random variable Y_i (conditioned on $R_{i-1}, \dots, R_1$) assumes two values with equal probability, and there exists $c_i \geqslant 0$ such that $Y_i \leqslant c_i$, then, for any $\lambda \geqslant 0$, we have*

$$\mathbf{Pr}[X_n - X_0 \geqslant \lambda] \leqslant \exp\left(-\frac{\lambda^2}{2\sum_{i=1}^n c_i^2}\right).$$

3 U_2-circuits

All known lower bounds for U_2-circuits [IM02, LR01, Zwi91, Sch74] were proved using the gate elimination method. We will generalize this method by defining a random process of restrictions under which the circuit size reduces with high probability. This allows us to get a #SAT algorithm for U_2-circuits of size almost $3n$, and also prove a correlation bound against Parity.

3.1 Concentrated shrinkage under restrictions

We call an $\wedge$-type function of two variables a *twig*. We now define a random process of restrictions where, at each step, we pick a variable or a twig and randomly assign it a value 0 or 1; we also simplify the circuit by eliminating unnecessary gates. The choice of variables or twigs at each step is determined by the following cases:

- If the circuit is a literal, choose the variable in the literal.
- If there is an input x with out-degree at least two, choose x.
- Otherwise, there must be a gate u fed by two variables having out-degree 1; we choose u (which is a twig).

Let C be a simplified U_2-circuit on inputs $x_1, \dots, x_n$. Let C' be the simplified circuit obtained after one step of restriction. Then we have the following lemma on the reduction of $\mu(C)$.

Lemma 3.1. *Suppose $\mu(C) \geqslant 4$. Let $\sigma = \mu(C) - \mu(C')$. Then we have $\sigma \geqslant 3$, and $\mathbf{E}[\sigma] \geqslant 4$.*

Proof. Consider the following cases (see also Figure 3.1):

(1) Suppose there is an input x_i feeding into two gates u and v. By Lemma 2.1, there is no edge between u and v. We randomly assign 0 or 1 to x_i, and consider the following sub-cases on the successors of u and v.

   (a) If u and v feed into two different successors, we have the following possibilities. If under one assignment to x_i, none of u, v become constants, then we can eliminate x_i, u, v; and under the other assignment to x_i, since both of u, v will be constants, we can eliminate two more gates (successors of u, v); thus we have $\mathbf{Pr}[\sigma \geqslant 5] \geqslant 1/2$, and $\sigma \geqslant 3$. If under each assignment to x_i, only one of u, v becomes a constant, then we can eliminate x_i, u, v and one successor; thus $\sigma \geqslant 4$.

   (b) If u and v feed into one single common successor w, we have similar situations as above. If under one assignment to x_i, both u and v become constants, then we can eliminate x_i, u, v, w and a successor of w; and under the other assignment to x_i, we can eliminate x_i, u, v. If under each assignment to x_i, only one of u, v becomes a constant, then we can eliminate x_i, u, v, w.

(2) If all inputs have out-degree 1, find a gate u fed by two inputs, say x_i and x_j. We randomly assign 0 and 1 to u; for each assignment, eliminate x_i, x_j, u and at least one successor of u. Then we have $\sigma \geqslant 4$.

In all cases, we have $\sigma \geqslant 3$, and $\mathbf{E}[\sigma] \geqslant 4$. $\square$

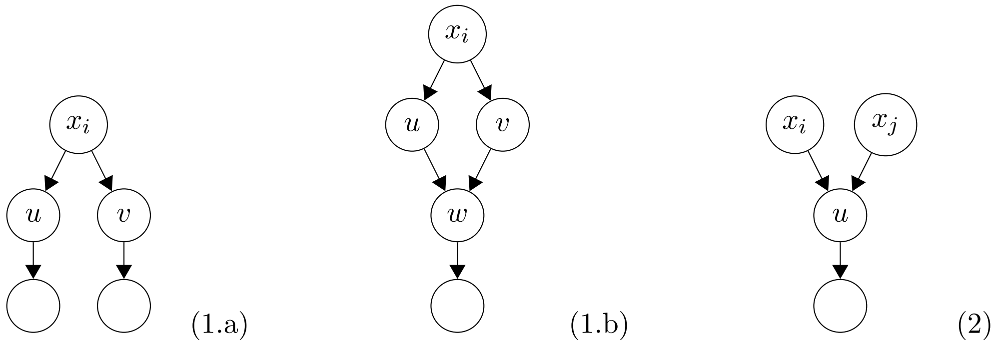

Figure 1: Cases in Lemma 3.1

Next consider the reduction of $\mu(C)$ under a sequence of restrictions. Let $C_0 := C$, and, for $i = 1, \ldots, d$, let C_i be the circuit obtained after the i-th step. For convenience, we let $\mu_i := \mu(C_i)$. Let R_i be the random value assigned to the variable or twig at each step. We define a sequence of random variables $\{Z_i\}$ as follows:

$$Z_i = \begin{cases} \mu_i - (\mu_{i-1} - 4), & \mu_{i-1} \geqslant 4, \\ 0, & \mu_{i-1} < 4. \end{cases}$$

Note that $0 < \mu_{i-1} < 4$ holds only when C_{i-1} itself is a literal or a twig, which means C_i will be a constant.

Lemma 3.2. *Let $X_0 = 0$ and $X_i = \sum_{j=1}^{i} Z_i$. Then we have $Z_i \leqslant 1$, and $\{X_i\}$ is a supermartingale with respect to $\{R_i\}$.*

Proof. By Lemma 3.1, conditioning on $R_1, \ldots, R_{i-1}$, when $\mu_{i-1} \geqslant 4$, we have $\mu_i \leqslant \mu_{i-1} - 3$ and $\mathbf{E}[\mu_i] \leqslant \mu_{i-1} - 4$. Therefore, we get $Z_i \leqslant 1$, $\mathbf{E}[Z_i \mid R_{i-1}, \ldots, R_1] \leqslant 0$, and $\mathbf{E}[X_i \mid R_{i-1}, \ldots, R_1] \leqslant X_{i-1}$. Thus $\{X_i\}$ is a supermartingale with respect to $\{R_i\}$. $\square$

Lemma 3.3. *For $\lambda \geqslant 0$, $\mathbf{Pr}\left[\mu_d \geqslant \max\{\mu_0 - 4d + \lambda, 1\}\right] \leqslant \exp(-\lambda^2/2d)$.*

Proof. Conditioning on $R_1, \ldots, R_{i-1}$, the variable Z_i assumes two values with equal probability. By Lemma 3.2, we have $\{X_i\}$ is a supermartingale with respect to $\{R_i\}$, and $Z_i \leqslant c_i \equiv 1$. Applying the bound in Lemma 2.4, we have

$$\mathbf{Pr}\left[\sum_{i=1}^{d} Z_i \geqslant \lambda\right] \leqslant \exp\left(-\frac{\lambda^2}{2d}\right).$$

When $\mu_d > 0$, we have $\sum_{i=1}^{d} Z_i = \mu_d - \mu_0 + 4d$. Let E_1 be the event that $\mu_d > 0$; let E_2 be the event that $\sum_{i=1}^{d} Z_i \geqslant \lambda$. Then the final probability is $\mathbf{Pr}[E_1 \wedge E_2] \leqslant \mathbf{Pr}[E_2] \leqslant \exp(-\lambda^2/2d)$. $\square$

3.2 #SAT algorithms

We now give a #SAT algorithm for circuits of size almost $3n$ based on the concentrated reduction of circuit size.

Theorem 3.4. *For U_2-circuits of size $s < 3n$, there is a deterministic #SAT algorithm running in time $2^{n-\Omega((3n-s)^2/n)}$.*

Proof. Let C be a circuit on n inputs $x_1, \ldots, x_n$ with size $s < 3n$. Let $\mu_0 := \mu(C) \leqslant s + n$.

We use the following procedure to construct a generalized decision tree, where each internal node is labeled by a variable or a twig. We start with the root node and C.

- If C is a constant, label the current node by this constant and return.
- Use the cases in Lemma 3.1 to find either a variable or a twig; denote it by u. Label the current node by u.
- Build two outgoing edges labeled by $u = 0$ and $u = 1$. For each child node, simplify the circuit, and recurse.

We say a complete assignment to $x_1, \ldots, x_n$ is *consistent* with a path (from the root to a leaf) if it satisfies the restrictions along the path. Since each assignment $a \in \{0,1\}^n$ is consistent with only one path, the paths give a disjoint partitioning of the boolean cube $\{0,1\}^n$. To count the number of satisfying assignments for C, one can count for each path with leaf labeled by 1, and return the summation. Restrictions along each path is essentially a read-once 2-CNF, for which counting is easy. We next only need to bound the size of the tree.

We wish to bound the probability that a random path has length larger than $n-k$, for k to be chosen later. Let $\lambda = 4(n-k) - \mu_0 + 1$. Then by Lemma 3.3, at depth $n-k$, the restricted circuit becomes a constant with probability at least $1 - \exp(-\lambda^2/2(n-k)) \geqslant 1 - 2^{-c\lambda^2/n}$ for a constant $c > 0$. The total number of paths with length larger than $n-k$ is at most

$$2^{n-k} \cdot 2^{-c\lambda^2/n} \cdot 2^k \leqslant 2^{n-c\lambda^2/n}.$$

Therefore, the size of the tree is at most $2^{n-k} + 2^{n-c\lambda^2/n}$. Choosing $k = (3n-s)/8$, both the tree size and the running time of the counting algorithm are bounded by $2^{n-\Omega((3n-s)^2/n)}$. $\square$

The following corollary is immediate.

Corollary 3.5. *(1) For U_2-circuits of size $3n - \epsilon n$ with $\epsilon > 0$, there is a deterministic #SAT algorithm running in time $2^{n-\Omega(n)}$. (2) For U_2-circuits of size $3n - n^\epsilon$ with $\epsilon > 0.5$, there is a deterministic #SAT algorithm running in time $2^{n-n^{\Omega(1)}}$.*

3.3 Correlation with Parity

Schnorr [Sch74] proved a $3n - c$ lower bound for computing Parity using the following fact: a simplified U_2-circuit computing Parity cannot have any input variable with out-degree exactly 1. Indeed, if such an input x exists, one can fix all other variables such that the gate fed by x becomes a constant, but this makes the function independent of x, which is impossible for Parity.

We next generalize this lower bound to the average case by showing that a U_2-circuit of size $s < 3n$ cannot approximate well with Parity. We will convert the generalized decision tree constructed in the proof of Theorem 3.4 into a decision tree without twigs, and argue that the tree size will not increase too much.

Lemma 3.6. *Any function computed by a U_2-circuit of size $s < 3n$ has a decision tree of size $2^{n-\Omega((3n-s)^2/n)}$.*

Proof. Let T be the (generalized) decision tree constructed in Theorem 3.4 for the given circuit. We expand each node labeled by a twig into two nodes labeled only by variables. For example, suppose we have a node labeled by $x \vee y$ with two subtrees A and B; we can replace it by two nodes x and y by making two copies of A. Denote the new decision tree by T', and we wish to bound the size of T'.

For a twig $x \vee y$, we say the restriction $x \vee y = 1$ is *good* (since it allows three configurations of x, y), whereas $x \vee y = 0$ is *bad*. We use similar definitions for the other types of twigs. For a path in T with l twigs having good restrictions, it will be replaced by 2^l paths in T'.

We first consider paths in T of length larger than $n - k$. As shown in Theorem 3.4, at depth $n - k$ of T, there are at most $2^{n-k} \cdot 2^{-c\lambda^2/n}$ nodes which are not leaves. Let v be such a node, and let l be the number of twigs on the path from the root to v. Then all paths in T passing through v will be replaced by at most $2^l \cdot 2^{k-l} = 2^k$ paths in T'. Therefore, all paths in T of length larger than $n - k$ will be replaced by at most $2^{n-c\lambda^2/n}$ paths.

For a path in T of length at most $n - k$, let l be the number of twigs with good restrictions along the path. If $l \leqslant k/2$, then this path is replaced by at most $2^{k/2}$ paths in T'. For all paths in T of length at most $n - k$ such that $l \leqslant k/2$, they will be replaced by at most $2^{n-k} \cdot 2^{k/2} = 2^{n-k/2}$ paths.

Consider a path of length at most $n - k$ which has $l > k/2$ twigs with good restrictions. After expanding the twigs, it is replaced by 2^l paths. When expanding a twig with a bad assignment, the path length increases by 1; when expanding a twig with a good assignment, the path becomes two paths with length increased by 0 and 1, respectively. Thus, by Chernoff bounds (choosing $\mu = l/2$ and $c = 1/2$ in Theorem 2.3), over the 2^l new paths, at most a fraction $2 \cdot e^{-l/32} < 2^{-k/c'}$ (for some constant c') will have length larger than $n - l + 3l/4 = n - l/4$. Therefore, there are at most $2^{n-k/c'}$ new paths having length larger than $n - k/8$.

Choosing $k = (3n - s)/8$ gives the result. □

The following lemma gives a simple relationship between the size of a decision tree and its correlation with Parity. It was previously used to derive correlation bounds for de Morgan formulas [San10] and AC^0 circuits [IMP12] .

Lemma 3.7. *A decision tree of size 2^{n-k} has correlation at most 2^{-k} with Parity.*

Proof. For a path of the decision tree with length strictly less than n, the restricted function is a constant, and thus it has zero correlation with Parity. Since there are less than 2^{n-k} paths with length exactly n, the decision tree computes Parity correctly on at most $1/2 + 2^{-k}$ fraction of all inputs. □

Theorem 3.8. *Let C be a U_2-circuit of size $s < 3n$. Then its correlation with Parity is at most $2^{-\Omega((3n-s)^2/n)}$. In particular, for $s = 3n - \epsilon n$ with $\epsilon > 0$, the correlation is at most $2^{-\Omega(n)}$; for $s = 3n - n^\epsilon$ with $\epsilon > 0.5$, the correlation is at most $2^{-n^{\Omega(1)}}$.*

Proof. The proof is immediate by Lemmas 3.6 and 3.7. □

The above correlation bounds with Parity almost match with the upper bounds. To see this, we can construction an approximate circuit for Parity in the following way. Divide n inputs into l

groups each of size n/l, use circuits of size $3(n/l-1)$ to compute Parity exactly for each group, and then take the disjunction of the outputs from all groups. This circuit outputs 0 with probability 2^{-l}, but whenever it outputs 0, it agrees with Parity. Thus its correlation with Parity is at least 2^{-l}. The circuit size is $3(n/l-1)\cdot l+l=3n-2l$.

Remark 3.9. The best U_2-circuit lower bound is $5n-o(n)$ [IM02, LR01]. It was proved against the so-called *strongly two-dependent* functions, which are functions such that fixing any two inputs results in four different sub-functions. Our approach cannot generalize this lower bound to the average case; a major difficulty is that an approximate circuit may not have the "strongly two-dependent" property.

3.4 Applications

Lemma 3.6 shows that, a circuit of size less than $3n$ has a decision tree of non-trivial size. Following from this property, one can get compression algorithms as in [CKK+14] and Fourier concentration result as in [IK14].

Corollary 3.10. *There is an algorithm running in time $2^{O(n)}$ such that, given the truth table of an (unknown) n-input boolean circuit of size $s<3n$, the algorithm produces an equivalent DNF of size $2^{n-\Omega((3n-s)^2/n)}\cdot \mathsf{poly}(n)$.*

This corollary follows directly from Lemma 3.6 and [CKK+14]. The decision tree constructed in Lemma 3.6 allows us to conclude that any function computed by U_2-circuits of size $s<3n$ has a DNF of size $S=n\cdot 2^{n-\Omega((3n-s)^2/n)}$. Then given the truth table, one can run a greedy set cover algorithm to construct an equivalent DNF of size at most $O(n)$ factor larger than S. We omit the proof.

Corollary 3.11. *Let f be a function computable by a boolean circuit of size $s<3n$. Then,*

$$\sum_{A\subseteq[n]:\,|A|>n-\Omega((3n-s)^2/n)} \widehat{f}(A)^2 \leqslant 2^{-\Omega((3n-s)^2/n)}.$$

This corollary follows from Lemma 3.6 and the fact that any decision tree of size S has $\sum_{A\subseteq[n]:\,|A|>k}\widehat{f}(A)^2\leqslant\epsilon$ for $k=\log(S/\epsilon)$ (see Proposition 3.17 in [O'D14]).

4 B_2-circuits

In this section, we give #SAT algorithms and correlation bounds for B_2-circuits of size almost $2.5n$.

4.1 Concentrated shrinkage and #SAT algorithms

Given a simplified B_2-circuit C, we will construct a generalized parity decision tree, where each internal node is labeled by either a twig or a parity of a subset of variables. Starting from the root with the given circuit C, we use the following case analysis to identify labels and build branches recursively.

If the circuit becomes a constant, we label the current node by the constant; then this node is a leaf. If the circuit is a literal or a gate fed by two variables, then we choose the variable of the literal or the circuit itself as the label, and build two branches. Otherwise, consider a topological

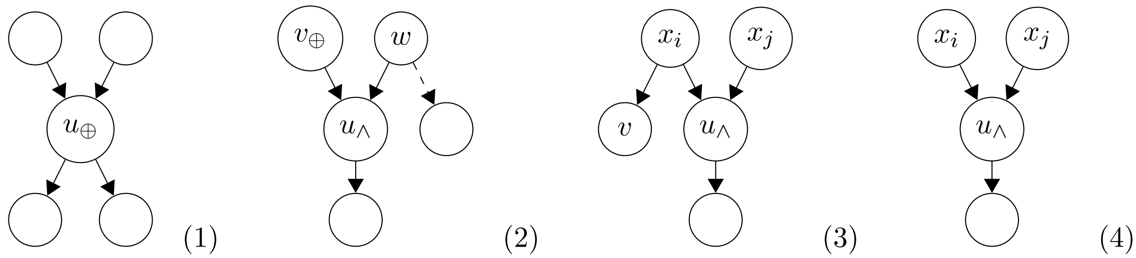

Figure 2: Cases for eliminating gates in B_2-circuits

order on the gates of the circuit, and let u be the first gate which is either $\oplus$-type of out-degree at least 2 or $\wedge$-type. Consider the following cases (see also Figure 4.1):

(1) If u is a $\oplus$-type gate of out-degree at least 2, then it computes $\oplus_{i\in I} x_i$ (or its negation) for some subset $I \subseteq [n]$. We choose $\oplus_{i\in I} x_i$ as the label, and build two branches; for the branch $\oplus_{i\in I} x_i = b \in \{0,1\}$, we replace u by a constant, and substitute an arbitrary variable x_j for $j \in I$ by a sub-circuit $\oplus_{i\in I\setminus\{j\}} x_i \oplus b$. In both branches, we can eliminate one variable x_j, and at least 3 gates (u and its two successors).

(2) If u is an $\wedge$-type gate fed by some $\oplus$-type gate v, suppose w is the other node feeding into u.

- If w has out-degree 1, then we choose the parity function computed at v as the label, and build two branches similar to Case (1). In one branch, we can eliminate some input x_j and two gates v, u; in the other branch, we can eliminate two more nodes: w and a successor of u.
- If w has out-degree at least 2, then it must be a variable. We choose w as the label, and build two branches. In one branch, we can eliminate w and its two successors; in the other branch, we can eliminate two more gates: v and a successor of u.

(3) If u is an $\wedge$-type gate fed by two inputs x_i and x_j where at least one of them, say x_i, has out-degree at least 2, then we choose x_i as the label and build two branches. In one branch, we can eliminate x_i and its two successors; in the other branch, we can eliminate one more gate: a successor of u.

(4) If u is an $\wedge$-type gate fed by two inputs each of out-degree 1, then choose the twig computed at u as the label. In both branches, we can eliminate x_i, x_j, u and a successor of u.

Consider a random path from the root of the decision tree to its leaves. Let $C_0 := C$, and let C_i be the restricted circuit obtained at depth i. Let $\mu_i := \mu(C_i)$. The next lemma follows directly from the above case analysis.

Lemma 4.1. *If $\mu_i > 4$, then $\mu_i - \mu_{i+1} \geqslant 3$, and $\mathbf{E}[\mu_i - \mu_{i+1}] \geqslant 3.5$. If $\mu_i \leqslant 4$, then $\mu_{i+1} = 0$.*

Then we have the following concentrated shrinkage.

Lemma 4.2. *For $\lambda \geqslant 0$, $\mathbf{Pr}\left[\mu_d \geqslant \max\{\mu_0 - 3.5d + \lambda, 1\}\right] \leqslant \exp(-\lambda^2/2d)$.*

Theorem 4.3. *For B_2-circuits of size $s < 2.5n$, there is a deterministic #SAT algorithm running in time $2^{n-\Omega((2.5n-s)^2/n)}$. In particular, for $s = 2.5n - \epsilon n$ with $\epsilon > 0$, the algorithm runs in time $2^{n-\Omega(n)}$; for $s = 2.5n - n^{\epsilon}$ with $\epsilon > 0.5$, the algorithm runs in time $2^{n-n^{\Omega(1)}}$.*

We omit the proofs of Lemma 4.2 and Theorem 4.3 since they are similar to the proofs of Lemma 3.3 and Theorem 3.4.

4.2 Correlation bounds

Demenkov and Kulikov [DK11] proved that affine dispersers for sources of dimension d requires B_2-circuits of size $3n - \Omega(d)$. We next extend this result to the average case by showing that affine extractors have small correlations with B_2-circuits of size less than $2.5n$.

Definition 4.4. Let F_2 be the finite field with elements $\{0, 1\}$. A function $\mathsf{AE}\colon F_2^n \to F_2$ is a (k, ϵ)-*affine extractor* if for any uniform distribution X over some k-dimensional affine subspace of F_2^n,

$$|\mathbf{Pr}[\mathsf{AE}(X) = 1] - 1/2| \leqslant \epsilon.$$

We will need the following constructions of affine extractors.

Theorem 4.5. *[Bou07, Yeh11, Li11] (1) For any $\delta > 0$ there exists a polynomial-time computable (k, ϵ)-affine extractor $\mathsf{AE}_1\colon \{0,1\}^n \to \{0,1\}$ with $k = \delta n$ and $\epsilon = 2^{-\Omega(n)}$. (2) There exists a constant $c > 0$ and a polynomial-time computable (k, ϵ)-affine extractor $\mathsf{AE}_2\colon \{0,1\}^n \to \{0,1\}$ with $k = cn/\sqrt{\log\log n}$ and $\epsilon = 2^{-n^{\Omega(1)}}$.*

We will prove our correlation bounds using the following representation of B_2-circuits by parity decision trees.

Lemma 4.6. *Any function computed by a B_2-circuit of size $s < 2.5n$ is computable by a parity decision tree of size $2^{n-\Omega((2.5n-s)^2/n)}$.*

The proof, which we omit here, is almost the same as the proof of Lemma 3.6. That is, using the algorithm in Theorem 4.3, one can construct a generalized parity decision tree which may have twigs, and then expand the twigs and argue that the tree size does not increase much. Note that, when we restrict a twig, the two variables in the twig are completely eliminated; when we restrict a parity, since one variable is substituted, all parity restrictions are linearly independent.

Lemma 4.7. *(1) For any $\delta > 0$, a parity decision tree of size 2^{n-k} for $k = \delta n$ has correlation at most $2^{-\Omega(n)}$ with AE_1. (2) There is a constant $c > 0$ such that a parity decision tree of size 2^{n-k} for $k = cn/\sqrt{\log\log n}$ has correlation at most $2^{-n^{\Omega(1)}}$ with AE_2.*

Proof. Consider a parity decision tree of size 2^{n-k} for $k = \delta n$. All paths from the root to leaves give a disjoint partitioning of the boolean cube $\{0, 1\}^n$.

For each path of length at most $n - k/2$, the inputs that are consistent with the path form an affine subspace of dimension at least $k/2$. Over all such short paths, by Theorem 4.5, the parity decision tree computes AE_1 correctly on at most $2^n \cdot (1/2 + 2^{-\Omega(n)})$ inputs. For paths of length larger than $n - k/2$, since the tree size is at most 2^{n-k}, the number of inputs that are consistent with these paths is at most $2^{n-k} \cdot 2^{k/2} = 2^{n-k/2}$. Therefore, the parity decision tree computes AE_1 correctly on at most a fraction $1/2 + 2^{-\Omega(n)} + 2^{-k/2} = 1/2 + 2^{-\Omega(n)}$ of the inputs.

The proof for the second case is similar. □

The next theorem follows by Lemma 4.6 and Lemma 4.7.

Theorem 4.8. *(1) For any $\delta > 0$ and B_2-circuit of size $2.5n - \delta n$, its correlation with AE_1 is at most $2^{-\Omega(n)}$. (2) There exists a constant $c > 0$ such that, for any B_2-circuit of size $2.5n - cn/\sqrt[4]{\log\log n}$, its correlation with AE_2 is at most $2^{-n^{\Omega(1)}}$.*

5 Open questions

It is open whether our correlation bounds (for the size almost $3n$ for U_2-circuits, and almost $2.5n$ for B_2-circuits) can be improved to match with the best known worst-case lower bounds (for the size almost $5n$ for U_2-circuits, and almost $3n$ for B_2-circuits). Pseudorandom generators for boolean formulas were constructed in [IMZ12] based on concentrated shrinkage and decomposition of the formula tree. It would be interesting to get pseudorandom generators for general boolean circuits.

References

[AB09] S. Arora and B. Barak. *Complexity theory: a modern approach.* Cambridge University Press, New York, 2009.

[BBC+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the Association for Computing Machinery*, 48(4):778–797, 2001.

[BIS12] P. Beame, R. Impagliazzo, and S. Srinivasan. Approximating ac^0 by small height decision trees and a deterministic algorithm for $\#ac^0$ sat. In *Proceedings of the 2012 IEEE Conference on Computational Complexity*, CCC '12, 2012.

[Blu84] N. Blum. A Boolean function requiring $3n$ network size. *Theoretical Computer Science*, 28:337–345, 1984.

[Bou07] J. Bourgain. On the construction of affine-source extractors. *Geometric and Functional Analysis*, 17(1):33–57, 2007.

[CKK+14] R. Chen, V. Kabanets, A. Kolokolova, R. Shaltiel, and D. Zuckerman. Mining circuit lower bound proofs for meta-algorithms. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity*, CCC '14, 2014.

[CKS14] R. Chen, V. Kabanets, and N. Saurabh. An improved deterministic #sat algorithm for small de morgan formulas. In *Proceedings of Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Part II*, pages 165–176, 2014.

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

[Hås86] J. Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 6–20, 1986.

[Hås98] J. Håstad. The shrinkage exponent of de Morgan formulae is 2. *SIAM Journal on Computing*, 27:48–64, 1998.

[Hås12] J. Håstad. On the correlation of parity and small-depth circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:137, 2012.

[IK14] R. Impagliazzo and V. Kabanets. Fourier concentration from shrinkage. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity*, CCC '14, 2014.

[IM02] K. Iwama and H. Morizumi. An explicit lower bound of $5n - o(n)$ for boolean circuits. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science*, MFCS '02, pages 353–364. Springer-Verlag, 2002.

[IMP12] R. Impagliazzo, W. Matthews, and R. Paturi. A satisfiability algorithm for AC^0. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 961–972, 2012.

[IMZ12] R. Impagliazzo, R. Meka, and D. Zuckerman. Pseudorandomness from shrinkage. In *Proceedings of the Fifty-Third Annual IEEE Symposium on Foundations of Computer Science*, pages 111–119, 2012.

[KR13] I. Komargodski and R. Raz. Average-case lower bounds for formula size. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, pages 171–180, 2013.

[KRT13] I. Komargodski, R. Raz, and A. Tal. Improved average-case lower bounds for demorgan formula size. In *Proceedings of the Fifty-Fourth Annual IEEE Symposium on Foundations of Computer Science*, pages 588–597, 2013.

[Li11] X. Li. A new approach to affine extractors and dispersers. In *IEEE Conference on Computational Complexity*, pages 137–147, 2011.

[LR01] O. Lachish and R. Raz. Explicit lower bound of $4.5n - o(n)$ for boolena circuits. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, STOC '01, pages 399–408, New York, NY, USA, 2001. ACM.

[Nur09] S. Nurk. An $o(2^{0.4058m})$ upper bound for circuit sat. *PDMI Preprint*, 2009.

[O'D14] R. O'Donnell. *Analysis of Boolean Functions.* Cambridge University Press, 2014.

[Rei11] B. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 560–569, 2011.

[San10] R. Santhanam. Fighting perebor: New and improved algorithms for formula and qbf satisfiability. In *Proceedings of the Fifty-First Annual IEEE Symposium on Foundations of Computer Science*, pages 183–192, 2010.

[Sch74] C. Schnorr. Zwei lineare untere schranken für die komplexität boolescher funktionen. *Computing*, 13(2):155–171, 1974.

[ST12] K. Seto and S. Tamaki. A satisfiability algorithm and average-case hardness for formulas over the full binary basis. In *Proceedings of the Twenty-Seventh Annual IEEE Conference on Computational Complexity*, pages 107–116, 2012.

[Sub61] B.A. Subbotovskaya. Realizations of linear functions by formulas using and, or, not. *Soviet Math. Doklady*, 2:110–112, 1961.

[Yao85] A.C. Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the Twenty-Sixth Annual IEEE Symposium on Foundations of Computer Science*, pages 1–10, 1985.

[Yeh11] A. Yehudayoff. Affine extractors over prime fields. *Combinatorica*, 31(2):245–256, 2011.

[Zwi91] U. Zwick. A 4n lower bound on the combinational complexity of certain symmetric boolean functions over the basis of unate dyadic boolean functions. *SIAM J. Comput.*, 20(3):499–505, 1991.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de