On ε-Biased Generators in NC^0

Elchanan Mossel *   Amir Shpilka†   Luca Trevisan‡

May 14, 2003

Abstract

Cryan and Miltersen [CM01] recently considered the question of whether there can be a pseudorandom generator in NC^0, that is, a pseudorandom generator that maps n bits strings to m bits strings and such that every bit of the output depends on a constant number k of bits of the seed. They show that for $k = 3$, if $m \geq 4n + 1$, there is a distinguisher; in fact,they show that in this case it is possible to break the generator with a *linear test*, that is, there is a subset of bits of the output whose XOR has a noticeable bias. They leave the question open for $k \geq 4$, and conjecture that every NC^0 generator can be broken by a statistical test that simply XORs some bits of the input. Equivalently, they conjecture that no NC^0 generator can sample an ε-biased space with negligible ε.

We refute the conjecture for $k \geq 5$, and we give a generator that maps n bits into cn bits, so that every bit of the output depends on 5 bits of the seed, and the XOR of every subset of the bits of the output has bias $2^{-\Omega(n/c^4)}$. For large values of k, we construct generators that map n bits to $n^{\Omega(\sqrt{k})}$ bits and such that every XOR of outputs has bias $2^{-n^{\frac{1}{2\sqrt{k}}}}$.

We also present a polynomial-time distinguisher for $k = 4, m \geq 24n$ having constant distinguishing probability. For large values of k we show that a linear distinguisher with a constant distinguishing probability exists once $m \geq \Omega(2^k n^{\lceil k/2 \rceil})$.

Finally, we consider a variant of the problem where each of the output bits is a degree k polynomial in the inputs. We show there exists a degree $k = 2$ pseudo random generator for which the XOR of every subset of the outputs has bias $2^{-\Omega(n)}$ and which map n bits to $\Omega(n^2)$ bits.

*`mossel@stat.berkeley.edu`. Miller fellow, CS and Statistics, U.C. Berkeley.

†`amirs@deas.harvard.edu`. DEAS, Harvard University and LCS, M.I.T., Cambridge, MA. Supported by National Security Agency (NSA) and Advanced Research and Development Activity (ARDA) under Research Office (ARO) contract no. DAAD19-01-1-0506.

‡`luca@cs.berkeley.edu`. Computer Science Division, U.C. Berkeley. Supported by a Sloan Fellowship and an Okawa Foundation Grant.

1 Introduction

A pseudorandom generator is an efficient deterministic procedure that maps a shorter random input into a longer output that is indistinguishable from the uniform distribution by resource-bounded observers.

A standard formalization of the above informal definition is to consider polynomial-time procedures G mapping n bits into $m(n) > n$ bits such that for every property P computable by a family of polynomial-size circuits we have that the quantity

$$\left| \mathop{\mathbf{Pr}}_{z\in\{0,1\}^{l(n)}}[P(z) = 1] - \mathop{\mathbf{Pr}}_{x\in\{0,1\}^n}[P(G(x))] \right|$$

goes to zero faster than any inverse polynomial in n. The existence of such a procedure G is equivalent to the existence of one-way functions [HILL99], pseudorandom functions [GGM86] and pseudorandom permutations [LR88].

What are the minimal computational requirements needed to compute a pseudorandom generator? Linial et al. [LMN93] prove that pseudorandom functions cannot be computed in AC^0 (constant-depth circuits with NOT gates and unbounded fan-in AND and OR gates),[1] but their result does not rule out the possibility that pseudorandom generators could be computed in AC^0, since the transformation of pseudorandom generators into pseudorandom functions does not preserve bounded-depth. Impagliazzo and Naor [IN96], in fact, present a candidate pseudorandom generator in AC^0. Goldreich [Gol00] suggests a candidate one-way function in NC^0. Recall that NC^0 is the class of functions computed by bounded-depth circuits with NOT gates and bounded fan-in AND and OR gates. In an NC^0 function, every bit of the output depends on a constant number of bits of the inputs. While it is easy to see that there can be no one-way function such that every bit of the output depends on only two bits of the input,[2] it still remains open whether there can be a one-way function such that every bit of the output depends on only three bits of the input.

Cryan and Miltersen [CM01] consider the question of whether there can be pseudorandom generators in NC^0, that is, whether there can be a pseudorandom generator such that every bit of the output depends only on some a constant k number of bits of the input.

They present a distinguisher in the case $k = 3, m > 4n$, and they observe that their distinguisher is a *linear* distinguisher, that is, it simply XORs a subset of the bits of the output. Cryan and Miltersen formulate a conjecture that implies that there is no pseudorandom generator in $\mathrm{NC}^0$when m is superlinear in n. Specifically, they conjecture that for every constant k if m is super-linear in n then for every generator such that every bit of the output depends on k bits of the input, a linear distinguisher exist. In order to formulate an equivalent version of the stronger conjecture, let us introduce the notion of a ε-*biased* distribution. For $\varepsilon > 0$, we say that a random variable $X = (X_1, \ldots, X_m)$ ranging over $\{0,1\}^m$ is ε-biased if for every subset $S \subseteq [m]$ we have $1/2 - \varepsilon \leq \mathbf{Pr}[\bigoplus_{i\in S} X_i = 0] \leq 1/2 + \varepsilon$. It is known [NN93, AGHP92] that an ε-biased distribution can be sampled by using only $O(\log(m/\varepsilon))$ random bits, which is tight up to the constant in the big-Oh. So the conjecture of [CM01] can be formulated as stating that there is no ε-biased generator in NC^0 that samples an m-bit ε-biased distribution starting from, say, $o(m)$ random bits and with a negligible ε.

[1] To be precise, the results in [LMN93] only rule out security against adversaries running in time $O(n^{(\log n)^{O(1)}})$.
[2] Finding an inverse can be formulated as a 2SAT problem.

Our Results

We first extend the result of Cryan and Miltersen by giving a (non linear) distinguisher for the case $k = 4, m \geq 24n$. Our distinguisher has a constant distinguishing probability, which we show to be impossible to achieve with linear distinguishers. Our distinguisher uses semidefinite programming and uses an idea similar to the "correlation attacks" used in practice against block cyphers.

For all k, it is trivial that a distinguisher exist for $m \geq 2^{2^k}\binom{n}{k}$, and it easy to see that a distinguisher exist when $m \geq k\binom{n}{k}$. We show using a duality Lemma proven in [MOS03] that in fact, a distinguisher with a constant distinguishing probability exists once $m \geq \Omega(2^k n^{\lceil k/2 \rceil})$.

Then we present an ε-biased generator mapping n bits into cn bits such that $\varepsilon = 1/2^{\Omega(n/c^4)}$ and every bit of the output depends only on $k = 5$ bits of the seed. The parameter c can be chosen arbitrarily, and may depend on n. The constant in the $\Omega()$ notation does not depend on c. The construction refutes the conjecture of Cryan and Miltersen.

The main idea in the construction is to develop a generator with $k = 3$ that handles well linear tests that XOR a *small* number of bits, and then develop a generator with $k = 2$ that handles well linear tests that XOR a *large* number of bits. The final generator outputs the bitwise XOR of the outputs of the two generators, on two independent seeds.

The generator uses a kind of unique-neighbor expander graphs that are shown to exist using the probabilistic method, but that are not known to be efficiently constructable, so the generator is in NC^0 but not in *uniform* NC^0.

Later we present similar constructions for large values of k which output $n^{\lfloor\sqrt{k}\rfloor\cdot(\frac{1}{2}+o(1))-2.5}$ bits whose bias is at most $\exp\left(-|\mathrm{n}|^{\frac{1-\mathrm{o}(1)}{2\lfloor\sqrt{\mathrm{k}}\rfloor}}\right)$.

Note the gap for large values of k between our constructions that output $n^{(\sqrt{k}/2)(1+o(1))}$ bits, and the bounds showing a distinguisher exists for generators that output $n^{(k/2)(1+o(1))}$ bits.

Finally, we begin a study of the question of whether there are pseudorandom generators with superlinear stretch such that each bit of the output is a function of the seed expressible as a degree-k polynomial over $GF(2)$, where k is a constant. This is a generalization of the main question addressed in this paper, since a function depending on only k inputs can always be expressed as a degree-k polynomial. Furthermore, low-degree polynomials are a standard class of "low complexity" functions from an algebraic perspective. In our $\mathrm{NC}^0{}_5$ construction of an ε-biased generator with exponentially small ε and superlinear stretch, every bit of the output is a degree-2 polynomial. We show that, for degree-2 polynomials, the stretch can be improved to quadratic, which is best possible.

Organization

In section 2 we review the analysis for the case $k = 3$ of [CM01]. In section 3 we give a distinguisher for the case $k = 4$. In section 4 we prove an upper bound on the length of the output of an ε-bias generator in NC^0_k.

In section 5 we construct ε-bias generator for the cases $k = 4, 5$. The results for larger k are discussed in section 6. In section 7 we explicitly construct an ε-bias generator such that every bit of the output is a polynomial of degree 2. Finally we give some open problems in section 8.

2 Review of the Case $k = 3$

In this section we summarize the main result of [CM01]. We first generalize a lemma from that paper.

2.1 Preliminaries

We say that a function $g : \{0,1\}^n \to \{0,1\}$ is *balanced* if $\mathbf{Pr}_x[g(x) = 1] = 1/2$. We say that a function $g : \{0,1\}^n \to \{0,1\}$ is *unbiased* towards a function $f : \{0,1\}^n \to \{0,1\}$ if $\mathbf{Pr}_x[g(x) = f(x)] = 1/2$.

Definition 1 (Affine function) *A function $g : \{0,1\}^n \to \{0,1\}$ is* affine *if there are values $a_0, \ldots, a_n \in \{0,1\}$ such that $g(x_1, \ldots, x_n) = a_0 \oplus a_1 x_1 \oplus \ldots \oplus a_n x_n$.*

The following lemma was proved by case analysis for $k = 3$ in [CM01], and the case $k = 4$ could also be derived from a case analysis appearing in [CM01] (but it is not explicitly stated).

Lemma 2 *Let $g : \{0,1\}^n \to \{0,1\}$ be a non-affine function that depends on only k variables. Then*

- *There exist an affine function on at most $k - 2$ variables that is correlated with g.*
- *Let l be the affine function that is biased towards g and that depends on a minimal number of variables. That is, for some d, l depends on d variables, $\mathbf{Pr}_x[g(x) = l(x)] > 1/2$, and g is unbiased towards affine functions that depend on less than d variables.*
  Then $\mathbf{Pr}_x[g(x) = l(x)] \geq 1/2 + 2^{d-k}$.

For example, for $k = 3$, a non-affine function g is either unbalanced, or it is biased towards one of its inputs; in the latter case it agrees with an input bit (or with its complement) with probability at least $3/4$.

For $k = 4$, a function g either is affine, or it is unbalanced, or it has agreement at least $5/8$ with an affine function that depends on only one input bit, or it has agreement at least $3/4$ with an affine functions that depends on only two input bits.

We give the proof of the lemma in appendix A.

2.2 The Case $k = 3$

Let $G : \{0,1\}^n \to \{0,1\}^m$ be a generator and let $g_i : \{0,1\}^n \to \{0,1\}$ be the i-th bit of the output of the generator. Suppose each g_i depends on only three bits of the input.

Suppose that one of the g_i is not a balanced function. Then we immediately have a distinguisher.

Suppose that more than n of the g_i are affine. Then one of them is linearly dependent of the others, and we also have a distinguisher.

It remains to consider the case where at least $m-n$ of the functions g_i are balanced and not affine. Let I be the set of i for which g_i is as above. Then, by Lemma 2, for each such g_i there is a affine function l_i that depends on only *one* bit, such that g_i agrees with l_i on a $3/4$ fraction of the inputs. By replacing g_i with $g_i \oplus 1$ when needed, we may assume that each such g_i has high correlation with one of the bits of its input. By the pigeonhole principle, there is a bit x_j of the seed, and a set C, $|C| \geq 1 + (m-n-1)/n$, such that the output of $g_i(x_1, \dots, x_n)$ is correlated to x_j for every $i \in C$. Let $c = |C|$. We see that the average over x of $\max\{\#i \in C : g_i(x) = 0, \#i \in C : g_i(x) = 1\}$ is at least $3c/4$. If c is a sufficiently large constant, then the restriction of the generator to C has constant statistical distance from the uniform distribution over c bits, for which that average value is $c/2 + O(\sqrt{c})$. By the Vazirani XOR Lemma [Vaz86], it also follows that the XOR of some subset of the bits of C has constant bias.[3]

While the above analysis uses the same ideas as in [CM01], it is slightly better because we achieve constant bias instead of inverse polynomial bias.

We state for future reference the following result that follows from the above analysis.

Lemma 3 *For every $\delta > 0$ there are constant $c_\delta = O(1/\delta^2)$ and $\varepsilon_\delta = 2^{-O(1/\delta^2)}$ such that the following holds. Let $G : \{0,1\}^n \to \{0,1\}^m$, and let $G(x) = (g_1(x), \dots, g_m(x))$. Let L be a set of functions and suppose that each function $g_i(x)$ agrees with an element of L or with its complement with probability at least $1/2 + \delta$, and that $m \geq 1 + c_\delta |L|$; then there is a set $C \subseteq [m]$ such that $\sum_{i \in C} g_i(x) \pmod 2$ has bias at least ε_δ.*

In particular, we can compute that when we flip 4 random coins, the average of the maximum between the number of zeroes and ones is $2.75 < \frac{3}{4} \cdot 4$, so we can set $c_{1/4} = 3$. In particular, we obtain a constant distinguishing probability once $m \geq 4n + 1$.

For the next section, it is useful to note that when we flip 10 random coins, the average of the maximum between the number of zeroes and ones is $6.23 < \frac{5}{8} \cdot 10$, so we can set $c_{1/8} = 9$.

3 Distinguisher for the Case $k = 4$

In this section we construct a distinguisher for $k = 4$.

Theorem 4 *Let $G = (g_1, \dots, g_m) : \{0,1\}^n \to \{0,1\}^m$ be a map such that each g_i depends on at most 4 coordinates of the input and $m \geq 24n$. Then there exists a polynomial time algorithm which distinguish between G and a random string with constant distinguishing probability. More precisely, the algorithm will output "yes" for the output of the generator G with probability $\Omega(1)$, and for a random string with probability $e^{-\Omega(m)}$.*

Note that it is easy to construct a distinguisher if any of the g_i is unbalanced, or if more than n of the g_i are linear.

If one of the g_i is biased towards one of the bits of its input, then it follows from Lemma 2 that it must agree with that bit or its complement with probability at least 5/8.

Thus, if more than $c_{1/8}n = 9n$ of the functions g_i have bias towards one bit, then we can obtain a distinguisher from Lemma 3.

It remains to consider the case where at least $m - 10n$ of the functions are balanced, non-linear, and unbiased towards single bits. Following [CM01], we call such functions *problematic*. It follows from Lemma 2 that for each problematic g there is an affine function l of two variables that agrees with g on a 3/4 fraction of the inputs. Again, by replacing g_i by $g_i \oplus 1$, when needed, we may assume that all the $g_i's$ in P have 3/4 agreement probability with some linear function.

[3] The Vazirani XOR Lemma is the fact that if $X_1, \dots, X_t$ are 0/1 random variables, then they are uniform and mutually independent if and only if for every non-empty $S \subseteq [t]$ we have $\mathbf{Pr}[\bigoplus_i X_i = 1] = 1/2$.

Let P be the set of i such that g_i is problematic. For each such i we denote by l_i the linear function of two inputs that agrees with g_i on a $3/4$ fraction of the inputs. In the next section we show how if $p = |P| \geq 14n$, one can "break" the generator using correlation attack. Correlation attacks are often used in practice to break pseudo random generators. The distinguisher below is a an interesting example where one can actually prove that correlation attack results in a polynomial time distinguisher.

3.1 The Distinguisher Based on Semidefinite Programming

Given a string $r_1, \ldots, r_p \in \{0,1\}^p$, consider the following linear system over $GF(2)$ with two variables per equation.

$$\forall i \in P \quad l_i(x) = r_i \tag{1}$$

We will argue that the largest fraction of satisfying assignments in the system (1) is distributed differently if $r_1, \ldots, r_p$ is uniform or if it is the output of G. By Markov inequality it follows that,

Lemma 5 *If $r_1, \ldots, r_p$ is the output of G, then, for every $\varepsilon > 0$, there is a probability at least ε that at least $3/4 - \varepsilon$ fraction of the equations in (1) are satisfiable. More formally*

$$\Pr_{z \in \{0,1\}^p} \left[\#\{\, i \mid g_i(z) = \ell_i(z) \} \geq \frac{3}{4} - \varepsilon \right] \geq \varepsilon$$

Lemma 6 *If $r_1, \ldots, r_p$ is chosen uniformly at random from $\{0,1\}^p$, and $|P| > (1/2\delta^2)(\ln 2)(n+c)$, then the probability that there is an assignment that satisfies more than a $1/2 + \delta$ fraction of the equations of (1) is at most 2^{-c}.*

Proof: Fix an assignment z; then the probability that a fraction at least $1/2 + \delta$ of the r_i agree with $l_i(z)$ is at most $e^{-2\delta^2 p} \leq 2^{-c-n}$. By a union bound, there is at most a probability 2^{-c} that such a z exists. □

Given a system of linear equations over $GF(2)$ with two variables per equation, it is NP-hard to determine the largest number of equations that can be satisfied, but the problem can be approximated to within a .878 factor using semidefinite programming [GW95]. We now prove Theorem 4

Proof of Theorem 4: Fix ε and δ small enough so that $.878(3/4 - \varepsilon) > 1/2 + \delta$. We now get a polynomial time algorithm that is successful if a fraction $3/4 - \varepsilon$ of the equations is holds, and fails if no more than $0.878(3/4 - \varepsilon)$ of the equations hold. Fixing $\delta = .158$ and $\varepsilon = 10^{-4}$, we obtain the statement of theorem, where $p = 14n$. □

3.2 Correlation Attacks

In this section we discuss how our distinguisher for the case $k = 4$can be seen as a "correlation attack."

Correlation attacks are a class of attacks that are often attempted in practice against candidate pseudorandom generators,[4] see e.g. the introduction of [JJ99] for an overview.

[4]Pseudorandom generators are called "block ciphers" in the applied cryptography literature.

The basic idea is as follows. Given a candidate generator $G : \{0,1\}^n \to \{0,1\}^m$, where $G(x) = g_1(x), \ldots, g_m(x)$, we first try and find linear relations between input bits and output bits that are satisfied with non-trivial probability. For example, suppose we find coefficients $a_{i,j}$, $b_{i,j}$ and c_j such that each of the equations

$$\begin{array}{l} \sum_{i=1}^n a_{i,1}x_i + \sum_{i=1}^m b_{i,1}g_i(x) = c_1 \pmod 2 \\ \sum_{i=1}^n a_{i,2}x_i + \sum_{i=1}^m b_{i,2}g_i(x) = c_2 \pmod 2 \\ \ldots \\ \sum_{i=1}^n a_{i,t}x_i + \sum_{i=1}^m b_{i,t}g_i(x) = c_t \pmod 2 \end{array} \tag{2}$$

is satisfied with probability bounded away from 1/2.

Now we want to use this system of equations in order to build a distinguisher. The distinguisher is given a sample $\mathbf{z} = (z_1, \ldots, z_m)$ and has to decide whether $\mathbf{z}$ is uniform or is the output of G. The distinguisher substitutes z_i in place of $g_i(x)$ in (2) and then tries to find an $\mathbf{x}$ that maximizes the number of satisfied equations. The hope is that, if $\mathbf{z} = G(\mathbf{x})$, then we will find $\mathbf{x}$ as a solution of the optimization problem.

Unfortunately, maximizing the number of satisfied equations in a linear system over $GF(2)$ is an NP-hard problem, and, in fact, it is NP-hard to achieve an approximation factor better than 1/2 [Hås97]. In practice, one uses belief-propagation algorithms that often work, although the method is typically not amenable to a formal analysis.

In Section 3, we were able to derive a formal analysis of a related method because we ended up with a system of equations having only two variables per equation, a class of instances for which good approximation algorithms are known. Furthermore, we did not try to argue that, when the method is applied to the output of the generator, we are likely to recover the seed; instead, we argued that just being able to approximate the largest fraction of satisfiable equations gives a way to distinguish samples of the generators from random strings.

4 $O(n^{k/2})$ upper bound

In this section we state the following theorem which gives an upper bound on the maximal stretch of an ε-bias generator in NC^0_k.

Theorem 7 *Let $\varepsilon > 0$, then there exists a constant c_ε such that if $G = (g_1, \ldots, g_m)$ is an ε biased pseudo random generator, where each of the g_i's depend on at most k bits, then $m \le c_\varepsilon 2^k n^{\lceil k/2 \rceil}$.*

For the proof we will utilize the following lemma from [MOS03].

Lemma 8 ([MOS03]) *Let $f : \{0,1\}^k \to \{0,1\}$ then for all r*

- *Either f is a degree r polynomial over F_2, or*
- *f is biased towards an affine function of $k - r$ variables.*

Proof of Theorem 7: Set $r = \lceil k/2 \rceil, s = k - r$ and $B_r = \sum_{i=0}^r \binom{n}{i}, B_s = \sum_{i=0}^s \binom{n}{i}$. Note that there exists a constant c such that $B_s \le B_r \le cn^{\lceil k/2 \rceil}$. By lemma 8 every g_i is either a degree $\le r$ polynomial, or is biased towards an affine function of at most s variables. Let p be the number of degree $\leq r$ polynomials and b be the number of functions biased towards an affine function of at most s variables. Clearly, $m \leq p + b$.

Note that the B_r monomials of degree $\leq r$ on the variables $x_1, \ldots, x_n$ form a basis to the vector space of all degree $\leq r$ polynomials in $x_1, \ldots, x_n$. Therefore if $p > t$, there is a linear dependency between the $g_i's$. We therefore conclude that

$$p \leq B_r \leq cn^{\lceil k/2 \rceil}. \tag{3}$$

On the other hand, note that by Lemma 2, if g is biased towards an affine function of at most s variables, then there exist an affine function ℓ of at most s variables such that $\mathbf{Pr}[f = \ell] \geq 1/2 + 2^{s-k} \geq 1/2 + 2^{-k/2}$. Moreover, there are exactly B_s linear functions on s variables.

Now Lemma 3 implies that there exists a constant $c'(\varepsilon)$ such that if $s \geq c'(\varepsilon) B_r 2^k$ then there is a $\oplus$ of a subset of the g_i's such that has an ε bias. It therefore follows that

$$b \leq c'(\varepsilon) 2^k B_r \leq cc'(\varepsilon) 2^k n^{\lceil k/2 \rceil}. \tag{4}$$

Combining (4) and (3) we obtain that

$$m \leq p + b \leq c(c'(\varepsilon)2^k + 1) n^{\lceil k/2 \rceil},$$

as needed. $\square$

5 Constructions for the Case $k = 5$ and $k = 4$

5.1 Preliminaries

We will construct a generator mapping $2n$ bits into cn bits; we think of c as an arbitrarily large constant (for every c, the construction is possible for every large enough n), although super-constant c is also achievable.

In fact, we will construct two generators: one will be good against linear tests that involve a small number of output bits (we call them *small tests*), and another is good against linear tests that involve a large number of output bits (we call them *large tests*). The final generator will be obtained by computing the two generators on independent seeds, and then XOR-ing their output bit by bit. In this way, we fool every possible test.

The generator that is good against large tests is such that every bit of the output is just the product of two bits of the seed. We argue that the sum (modulo 2) of t output bits of the generator has bias exponentially small in t/c^2, where c, as above, is the stretch of the generator.

Then we describe a generator that completely fools linear tests of size up to about n/c^2, and such that every bit of the output is the sum of three bits of the seed. Combined with the generator for large tests, we get a generator in NC^0_5 such that every linear test has bias $2^{-O(n/c^4)}$.

5.2 The Generator for Large Tests

Let us call the bits of the seed $y_1, \ldots, y_n$.

Let K be an undirected graph formed by $n/(2c+1)$ disjoint cliques each with $2c+1$ vertices. Then K has n vertices, that we identify with the elements of $[n]$, and $cn = m$ edges. Fix some ordering of the edges of K, and let (a_j, b_j) be the j-th edge of K. Define the functions $q_1, \ldots, q_m$ as $q_j(y_1, \ldots, y_n) = y_{a_j} y_{b_j}$.

Claim 9 *For every subset $S \subset [m]$, the function $q_S(\mathbf{y}) = \sum_{j \in S} q_j(\mathbf{y})$ is such that*

$$\frac{1}{2} - \left(\frac{1}{2}\right)^{1+|S|/(2c^2+c)} \leq \mathbf{Pr}_{\mathbf{y}}[q_S(\mathbf{y}) = 0] \leq \frac{1}{2} + \left(\frac{1}{2}\right)^{1+|S|/(2c^2+c)}$$

The proof relies on the following two lemmas. The first one is from [CM01], and it is easy to prove it by induction on the number of variables, and the second one is standard and it is easy to prove it by replacing $\{0,1\}$ with $\{-1,1\}$ and $\oplus$ with multiplication.

Lemma 10 ([CM01]) *Let p be a non-constant degree-2 multilinear polynomial over $GF(2)$. Then $1/4 \leq \mathbf{Pr}[p(x) = 0] \leq 3/4$.*

Lemma 11 *Let $X_1, \ldots, X_t$ be independent 0/1 random variables, and suppose that for every i we have $\delta \leq \mathbf{Pr}[X_i = 0] \leq 1 - \delta$. Then*

$$\frac{1}{2} + \frac{1}{2}(1-2\delta)^t \leq \mathbf{Pr}\left[\bigoplus_i X_i = 0\right] \leq \frac{1}{2} + \frac{1}{2}(1-2\delta)^t$$

We can now prove Claim 9.

Proof of Claim 9. We can see S as a subset of the edges of K. Each connected component of K has $2c^2 + c$ edges, so S contains edges coming from at least $|S|/(2c^2+c)$ different connected components, let us call t this number. If we decompose the summation $\sum_{j \in S} q_j(y_1, \ldots, y_n)$ into terms depending on each of the connected components, then each term is a non-trivial degree-2 polynomial, and the t terms are independent random variables when $y_1, \ldots, y_n$ are picked at random. We can then apply Lemma 11, where the X_i are the values taken by each of the t terms in the summation, $\delta = 1/4$, and $t = |S|/(2c^2+c)$. $\square$

5.3 The Generator for Small Tests

Let $A \in \{0,1\}^{n \times m}$ be a matrix such that every row is a vector in $\{0,1\}^n$ with exactly three non-zero entries, and let also A be such that every subset of σ rows are linearly independent. Let $A_1, \ldots, A_m$ be the rows of A. We define the linear functions $l_1, \ldots, l_m$ as $l_i(\mathbf{x}) = A_i \cdot \mathbf{x}$. Note that each of these linear functions depends on only three bits of the input.

Claim 12 *For every subset $S \leq [m]$, $|S| < \sigma$, the function $l_S(\mathbf{x}) = \sum_{j \in S} l_j(\mathbf{x})$ is balanced.*

Proof: We have $l_S(\mathbf{x}) = (\sum_{j \in S} A_j) \cdot \mathbf{x}$, and since $\sum_{j \in S} A_j$ is a non-zero element of $\{0,1\}^n$, it follows that $l_S()$ is a non-trivial linear function, and therefore it is balanced. $\square$

There are matrices with linear σ.

Lemma 13 *For every $c = c(n) = o(\sqrt{n}/(\log n)^{3/4})$ and for sufficiently large n there is a 0/1 matrix A with cn rows and n*

columns such that every row has exactly three non-zero entries and such that every subset of $n/(4e^2c^2(n))$ rows are linearly independent.

This is a standard probabilistic construction Similar calculations have been done several times, for example in [BKPS98, BSW01, BOT02]. We give the calculation in Appendix B for the sake of self-containment.

5.4 Putting Everything Together

In order to obtain the generator, we take $G_1 : \{0,1\}^n \to \{0,1\}^m$ to be a generator satisfying Claim 9, and $G_2 : \{0,1\}^n \to \{0,1\}^m$ to satisfy Lemma 13. Then we take $G : \{0,1\}^{2n} \to \{0,1\}^m$ defined by $G(x,y) = G_1(x) \oplus G_2(y)$ to fool both small tests and large tests. We thus obtain

Theorem 14 *For every c and sufficiently large n, there is a generator in NC^0_5 mapping n bits into cn bits and sampling an ε-biased distribution, where $\varepsilon = 2^{-n/O(c^4)}$.*

5.5 Generator for the case $k = 4$

When $k = 4$ we want to replace the generator for small sets by a generator which depends only on two bits. The construction is essentially the one in [CM01].

The generator is obtained by taking a graph H on cn edges, with girth $\Omega(\log n/\log c)$ and letting $x_i \oplus x_j$ be an output bit, if (i,j) is an edge of the graph.

Let H be an undirected graph with n vertices, that we identify with $[n]$, having cn edges and girth γ. Fix some ordering of the edges of H, and let (a_j, b_j) be the j-th edge of H. We define the linear functions $l_1, \ldots, l_m$ as $l_i(x_1, \ldots, x_n) = x_{a_j} + x_{b_j}$.

Claim 15 *For every subset $S \leq [m]$, $|S| < \gamma$, the function $l_S(\mathbf{x}) = \sum_{j\in S} l_j(\mathbf{x})$ is balanced.*

PROOF: We can see S as a set of edges in H, and l_S as the function that sums x_i for each vertex i that is incident on an odd number of edges in S. Since $|S| < g$, the subgraph of H induced by the edges of S is a forest, and so some vertex must have odd degree (in fact, some vertex must have degree one). It follows that l_S is the sum of a non-empty subset of its inputs, and so it is balanced.[5] □

We can let γ be as large as about $\log_c n$.

Lemma 16 ([LPS88]) *For every c and for sufficiently large n there are explicitly constructible graphs H with n vertices, cn edges, and girth $\Omega((\log n)/(\log c))$.*

We thus obtain.

Theorem 17 *For every c and sufficiently large n, there is a generator in uniform NC^0_4 mapping n bits into cn bits and sampling an ε-biased distribution, where $\varepsilon = n^{-1/O(c^2 \log c)}$.*

[5] Equivalently, we proved that every subset of $< \gamma$ of the functions l_i are linearly independent.

6 ε-biased generator for large k

In this section we construct an ε-biased generator in NC^0_k, for large k, which outputs $n^{\Omega(\sqrt{k})}$ bits. More precisely,

Theorem 18 *Let k be a positive integer. There exist an ε-bias generator in NC^0_k from n bits to $n^{\lfloor\sqrt{k}\rfloor\cdot(\frac{1}{2}+o(1))-2.5}$ bits whose bias is at most*

$$\exp\left(-|\mathrm{n}|^{\frac{1-o(1)}{2\lfloor\sqrt{k}\rfloor}}\right)$$

Proof: Let $k' = (\lfloor\sqrt{k}\rfloor - 5)^2$, $n' = \lfloor\sqrt{\frac{n}{2}}\rfloor^2$. We have that

$$k > k' + 10\sqrt{k'}$$

$$k' > k - 12\sqrt{k}$$

$$\frac{n}{2} \geq n' > \frac{n}{2} - \sqrt{2n}$$

Let $X = \{x_1, ..., x_{n'}\}$, $Y = \{y_1, ..., y'_n\}$. Let $f_1(X), \ldots, f_{\binom{m}{d}}(X)$ be the outputs of the generator against long tests with the parameters $m = \sqrt{n'}$, $d = \sqrt{k'}$. Let $h_1(Y), \ldots, h_{n'^{k'}}(Y)$ be the outputs of the generator for small tests on Y, given the parameter $t = \sqrt{k'}$. Note that

$$n'^{k'} > \binom{\sqrt{n'}}{\sqrt{k'}} = \binom{m}{d}$$

Our generator G will output the functions

$$\forall 1 \leq i \leq \binom{m}{d} \quad g_i(X,Y) = f_i(X) + h_i(Y)$$

Notice that as we have more h_i's than f_i's we don't use most of the h_i's. Clearly, each output of the generator depends on $k' + 10\sqrt{k'} < k$ input variables.

From lemma 20,23 we get that the bias of any non trivial linear combination of the outputs is at most

$$\exp\left(\frac{-|\mathrm{n}'|^{\frac{1}{2\mathrm{d}}}}{2^{\mathrm{d}}}\right)$$

Thus our generator takes $2n' \leq n$ inputs and outputs

$$\binom{m}{d} \geq \left(\frac{e^2 n'}{k'}\right)^{\frac{\sqrt{k'}}{2}} = n^{\lfloor\sqrt{k}\rfloor\cdot(\frac{1}{2}+o(1))-2.5}$$

and has an exponentially small bias. $\square$

6.1 The Generator for Large Tests

We introduce new parameters which will simplify the presentation of the construction. Let $d \approx \sqrt{k}$, $m \approx \sqrt{n}$.

Consider the following bi-partite graph $G = (L, R, E)$ where $|L| = m$, $|R| = \binom{m}{d}$. Identify the vertices of L with the numbers $1, ..., m$ and the vertices of R with $\binom{[m]}{d}$, the set of all subsets of $[m]$ of size d. The edges of G are all pairs (i, S) such that $i \in [m]$, $S \in \binom{[m]}{d}$ and $i \in S$.

For a set of vertices, V, we denote with $N(V)$ the set of neighbors of V. For a vertex i let $\mathrm{deg(i)} = |\mathrm{N(i)}|$.

The following claim is obvious

Claim 19 *For any set of right vertices $V \subset R$ we have that $|N(V)| \geq \frac{d|V|^{\frac{1}{d}}}{e}$.*

Proof: Any set of t left vertices has $\binom{t}{d}$ right neighbors. The result follows from the inequality

$$|V| \leq \binom{|N(V)|}{d} \leq \left(\frac{e|N(V)|}{d}\right)^d$$

$\square$

Our construction will assign a monomial of degree d, in the input variables, to each edge. We think about the vertices of L as representing disjoint subsets of the input variables and each edge leaving such input set corresponds to a monomial in its variables. The right vertices, R, correspond to the output bits. Each output is the sum of monomials that label the edges that fan into it. We now give the formal construction.

Let $X = \bigsqcup_{i=1}^m X_i$ be a partition of $X = \{x_1, ..., x_n\}$ to m disjoint sets each of size m.[6]

We assign the set X_i to the i'th vertex of L. Let M_i be the set of all multilinear monomials of degree d in the variables of X_i. We have that

$$|M_i| = \binom{m}{d} > \binom{m-1}{d-1} = \mathrm{deg(i)}$$

Therefor we can assign to each edge leaving i a different monomial from M_i.

Each right vertex corresponds to an output bit. For a right vertex j the j'th output is the sum of all monomials that were assigned to the edges adjacent to j. Thus each output is the sum of d monomials each of degree d. Hence each output depends on d^2 input variables. Denote with f_j the j'th output. We now show that any large linear combination has a small bias.

Lemma 20 *In the notations above any linear combination (over $GF(2)$) $f = \sum_{j \in J} f_j$ has bias at most*

$$\exp\left(\frac{-|\mathrm{J}|^{\frac{1}{\mathrm{d}}}}{2^{\mathrm{d}}}\right)$$

Proof: The proof is essentially the same as the proof of claim 9 and follows from the following easy claims.

[6] We assume for simplicity that $n = m^2$, otherwise we take $m = \lfloor \sqrt{n} \rfloor$ and $n' = m^2$. Since $n' \geq n - 2\sqrt{n}$ our results will not change much if we consider n' instead of n.

Claim 21 *f can be written as the sum of at least $N(J)$ polynomials of degree d, each in a different set of variables.*

PROOF: The set of outputs J, has $N(J)$ left neighbors. The edges connecting the set J to a neighbor $i \in N(J)$ are labeled with polynomials of degree d in X_i. $\square$

From the Schwartz-Zippel lemma [Sch80, Zip79] we get

Claim 22 *The bias of any polynomial of degree d is bounded above by $\frac{1}{2^d}$.*

Thus according to Lemma 11 we get that the bias of f is at most

$$\frac{1}{2}\left(1-\frac{2}{2^d}\right)^{N(J)} \leq \frac{1}{2}\cdot\exp\left(\frac{-2\mathrm{N(J)}}{2^{\mathrm{d}}}\right) \leq \exp\left(\frac{-|\mathrm{J}|^{\frac{1}{\mathrm{d}}}}{2^{\mathrm{d}}}\right)$$

This finishes the proof of lemma 20 $\square$

This finishes the construction of the generator for large tests. We now describe the generator for small tests.

6.2 The Generator for Small Tests

Similar to the $k = 4, 5$ cases this generator will output only linear functions. We will have the property that any small set of these linear functions is linearly independent. This kind of construction is standard and follows from unique neighbor property of expanding graphs.

The formulation we need is proven in appendix B.

Lemma 23 *Let t be positive integer t and $\Delta = 10t$. There exist a mapping from n bits to n^t bits such that every output depends on Δ input variables, and such that any linear combination of at most $\sqrt{n}$ outputs is linearly independent.*

7 A degree 2 generator

In this section we consider a variant of the problem presnted in the paper. Suppose that we require that every output bit is a degree k polynomial in the input bits. It is clear that if we want the output to be ε-biased, then the number of output bits m is at most the dimension of degree k polynomials in n variables $\sum_{i=k}^{s}\binom{n}{i} = O(n^k)$.

Clearly this is a relaxation of the problem described above. In particular any upper bound here will imply an upper bound for NC_k^0. The problem is also of independent interest, as low degree generators are "simple" in an intuitive sense.

In this section we construct a generator of ε-biased set such that every output is a polynomial of degree 2 in the input variables. We show that unlike the $k = 2$ case we can output $\Omega(n^2)$ bits. In particular we prove

Theorem 24 *For every $1 \leq m \leq n$ there exists an ε-bias generator $G = (g_1, ..., g_t) : \{0,1\}^n \mapsto \{0,1\}^t$, $t = \lfloor \frac{n}{2} \rfloor \cdot m$, such that g_i is a degree 2 polynomial, and the bias of any non trivial linear combination of the g_i's is at most $2^{\frac{n-2m}{4}}$.*

We begin by studying the bias of a degree 2 polynomial, over $GF(2)$.

7.1 The Bias of Degree 2 polynomials

Let $P(x_1, ..., x_n)$ be a degree 2 polynomial. P is also called a quadratic form over $GF(2)$. We say that a matrix A represents P with respect to a basis of $GF(2)^n$, $\{v_i\}_{i=1}^n$, if for every vector $v = \sum_{i=1}^n x_i \cdot v_i$ we have that

$$P(v) = x^t Ax$$

$(x = (x_1, ..., x_n))$. Notice that we can always find an upper triangular matrix that represents P; let

$$P(a_1, ..., a_n) = \sum_{1 \le i \le j \le n} \alpha_{i,j} a_i a_j$$

Define

$$A(P)_{i,j} = \begin{cases} \alpha_{i,j} & i \le j \\ 0 & i > j \end{cases}$$

Clearly $P(\sum_{i=1}^n e_i \cdot x_i) = x^t A(P) x$ and $A(P)$ represents P with respect to the standard basis.

There is a relation between the rank of a quadratic form and the rank of a matrix that represents it.

Theorem 25 *The bias of a degree 2 polynomial P is at most*

$$2^{-\left(1+\frac{\mathrm{rank(A+A^t)}}{4}\right)}$$

for any matrix A that represents P.

We give the proof in appendix 7.2. Theorem 25 shows that in order to output m polynomials of degree 2, such that any non trivial linear combination of them is almost unbiased we need to find matrices $A_1, ..., A_m$ such that for any non trivial combination of them, $B = \sum_{i=1}^m \alpha_i A_i$ $(\alpha_i \in GF(2))$, we have that $\mathrm{rank(B + B^t)}$ is high.

7.2 Proof of theorem 25

The following claim is trivial.

Claim 26 *$P \equiv 0$ iff there exist a symmetric matrix that represents P w.r.t. some basis iff any matrix that represents P is symmetric.*

The proof of theorem 25 will follow from the following claims.

Claim 27 *For any quadratic form P on n variables, there exist a basis of $GF(2)^n$ e_i, f_i $i = 1, ..., r$ and g_j $j = 1, ..., s$ such that $2r + s = n$ and n elements in $GF(2)$, a_i, b_i $i = 1, ..., r$, c_j $j = 1, ..., s$, such that for*

$$v = \sum_{i=1}^r x_i e_i + \sum_{i=1}^r x_{r+i} f_i + \sum_{j=1}^s x_{2r+j} g_j$$

we have

$$P(v) = \sum_{i=1}^r (a_i {x_i}^2 + x_i x_{r+i} + b_i {x_{r+i}}^2) + \sum_{j=1}^s c_j {x_{2r+j}}^2$$

Such a basis is called "a canonical basis for P".

PROOF: See the proof of theorem 5.1.7 in [Hir79]. □

Claim 28 *Let P be a quadratic form on n variables. Let A represent P with respect to the standard basis and D represent P with respect to the canonical basis. Then*

$$\mathrm{rank(D)} \geq \frac{\mathrm{rank(A + A^t)}}{2}$$

PROOF: Let B be the matrix whose columns are $e_1, ..., e_r, f_1, ..., f_r, g_1, ..., g_s$ written w.r.t. the standard basis. We have that

$$\forall x \in GF(2)^n \ \ x^t D x = x^t B^t A B x$$

In other words

$$\forall x \in GF(2)^n \ \ x^t (D - B^t A B) x = 0$$

Therefor there exist a symmetric matrix S such that

$$D - B^t A B = S$$

or

$$D = B^t (A + (B^{-1})^t S (B^{-1})) B$$

As $(B^{-1})^t S (B^{-1})$ is a symmetric matrix we get by the next claim (claim 29) that

$$\mathrm{rank(D)} \geq \frac{\mathrm{rank(A + A^t)}}{2}$$

□

Claim 29 *For upper diagonal matrix A with zeros on the diagonal, and any symmetric matrix S we have that*

$$\mathrm{rank(A + S)} \geq \frac{\mathrm{rank(A + A^t)}}{2}$$

where A^t is the transpose of A.

PROOF: Let $r = \mathrm{rank(A + S)}$. As $S = S^t$ we get

$$\mathrm{rank(A^t + S) = rank(A^t + S^t) = rank((A + S)^t) = rank(A + S) = r}$$

Hence

$$\mathrm{rank(A + A^t) \leq rank(A + S) + rank(A^t + S) = 2r}$$

□

PROOF OF THEOREM 25. Clearly the bias of P does not change if we calculate it w.r.t. to a canonical basis, $\{v_i\}_{i=1}^n$, for P. In such a basis we have that

$$P(\sum_{i=1}^{n} v_i \cdot x_i) = \sum_{i=1}^{r} (a_i {x_i}^2 + x_i x_{r+i} + b_i {x_{r+i}}^2) + \sum_{j=1}^{s} c_j {x_{2r+j}}^2$$

First notice that if for some $1 \leq j \leq s$ $c_j \neq 0$ then P is unbiased. Otherwise we note that for every i the bias of $(a_i {x_i}^2 + x_i x_{r+i} + b_i {x_{r+i}}^2)$ is at most $\frac{1}{4}$. Therefor according to lemma 11 we get the bias of P is at most $\left(\frac{1}{2}\right)^{r+1}$. As we assume that $\forall j \ \ c_j = 0$ we see that

$$r \geq \frac{\mathrm{rank(D)}}{2}$$

The theorem now follows from claim 28. □

7.3 Linear Space of Matrices of High Rank

In this subsection we construct a linear space of matrices with the property that for every non zero matrix in the space, A, we have that $\mathrm{rank}(A + A^t)$ is high.

Such a construction was first given by Roth [Roth91], and later simplified by Meshulam [Mes95]. The construction that we give here is taken from [Shp02] and is similar to the one in [Mes95].

Theorem 30 *For any positive natural numbers $n \geq m$ there exist $t = \lfloor \frac{n}{2} \rfloor \cdot m$ matrices $A_1, ..., A_t \in M_n(GF(2)$ such that for every non trivial combination of them $B = \sum_{i=1}^{t} \alpha_i A_i$ we have that*

$$\mathrm{rank}(B + B^t) \geq n - 2m$$

By combining theorem 25 and theorem 30 we can construct our generator.

The Generator

Theorem 31 *For every $1 \leq m \leq n$ there exists an ε-bias generator $G = (g_1, ..., g_n) : \{0,1\}^n \mapsto \{0,1\}^t$, $t = \lfloor \frac{n}{2} \rfloor \cdot m$, such that g_i is a degree 2 polynomial, and the bias of any non trivial linear combination of the g_i's is at most $2^{\frac{n-2m}{4}}$.*

Proof: Let $A_1, ..., A_t$ be the matrices guaranteed by theorem 30. Define $g_i(x) = x^t A_i x$. Consider any non trivial linear combination

$$g(x) = \sum_{i=1}^{t} \alpha_i g_i(x) = x^t \left(\sum_{i=1}^{n} \alpha_i A_i \right) x$$

According to theorem 30, we have that $\mathrm{rank}(g) \geq n - 2m$. Theorem 25 shows that the bias of g is at most $2^{\frac{n-2m}{4}}$. □

7.4 Proof of theorem 30

Denote with $\mathbb{F} = GF(2^n)$ the field with 2^n elements. $\mathbb{F}$ is a linear space over $GF(2)$ of dimension n. We will abuse notation and think about each $y \in \mathbb{F}$ both as a field element and as a vector in $GF(2)^n$. Fix a basis to $\mathbb{F}$ over $GF(2)$ of the form $1, x, x^2, ..., x^{n-1}$ for some $x \in \mathbb{F}$. Each element, $y \in \mathbb{F}$, can be viewed as a linear transformation of $\mathbb{F}$ over $GF(2)$ in the following manner:

$$\forall z \in \mathbb{F} \;\; y(z) = y \cdot z \; .$$

Thus for every $y \in \mathbb{F}$ there is a corresponding matrix $A_y \in M_n(GF(2))$, that represents y over the basis we chose. We denote $A = A_x$ (the same x as in the basis).

Let $\varphi : \mathbb{F} \mapsto \mathbb{F}$ be the Frobenius transformation, that is $\varphi(y) = y^2$. Let $\varphi^{(k)} = \varphi \circ \varphi ... \circ \varphi$, k times. That is $\varphi^{(k)}(y) = y^{2^k}$. It is easy to see that φ is a linear transformation of $\mathbb{F}$ over $GF(2)$. We denote with B the matrix that represents φ over our basis. That is, by abusing notations,

$$\forall y \in \mathbb{F} \;\; By = y^2$$

Let $V \subset M_n(GF(2))$ be the linear space spanned by the matrices

$$V = \mathrm{span}\{ \; A^i \cdot B^j \mid i = 0, ..., n-1 \; , \; j = 0, ..., m-1 \; \}$$

Lemma 32 *V is a linear space of matrices of dimension nm such that for any $0 \neq E \in V$ we have that*

$$\mathrm{rank(E) > n - m}$$

PROOF: Let $0 \neq E \in V$. We want to calculate $dim(ker(E))$. For any $y \in \mathbb{F}$ we think about Ey also as an element of $GF(2)^n$. It is clear that

$$Ey = \left(\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \alpha_{i,j} A^i \cdot B^j \right) y = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \alpha_{i,j} A^i (y^{2^j}) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \alpha_{i,j} x^i y^{2^j} .$$

That is, Ey is a polynomial of degree 2^{m-1} in y. Therefore it has at most 2^{m-1} roots. As E is a linear transformation, we get that its roots are a linear space. Since there are at most 2^{m-1} roots, the dimension of $ker(E)$ is at most $m-1$. Hence $rank(E) \geq n - m + 1$. $\square$

PROOF OF THEOREM 30 Let V be the space guaranteed by lemma 32 in $M_{\lfloor \frac{n}{2} \rfloor}(GF(2))$ of dimension $t = \lfloor \frac{n}{2} \rfloor \cdot m$ Let $E_1, ..., E_t$ be a basis for V. Let A_i be a $n \times n$ matrix of the following form

$$A_i = \begin{pmatrix} 0 & E_i \\ 0 & 0 \end{pmatrix}$$

Where the 0 stands for the all zero matrix in $M_{\lfloor \frac{n}{2} \rfloor}(GF(2))$. For any non trivial combination $B = \sum_{i=1}^{t} \alpha_i A_i$ we get

$$B = \sum_{i=1}^{t} \alpha_i A_i = \begin{pmatrix} 0 & \sum_{i=1}^{t} \alpha_i E_i \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & E \\ 0 & 0 \end{pmatrix}$$

where $E = \sum_{i=1}^{t} \alpha_i E_i$. Since $\{E_i\}$ is a basis and not all the α_i's are zero then $0 \neq E \in V$. Therefore $\mathrm{rank(E)} \geq \lfloor \frac{\mathrm{n}}{2} \rfloor - \mathrm{m} + 1$. We get that

$$\mathrm{rank(B + B^t) = rank} \begin{pmatrix} 0 & E \\ E & 0 \end{pmatrix} = 2 \cdot \mathrm{rank(E)} \geq 2(\lfloor \frac{\mathrm{n}}{2} \rfloor - \mathrm{m} + 1) \geq \mathrm{n} - 2\mathrm{m}$$

$\square$

8 Conclusions

Several questions remain open.

Even for the case $k = 3$, we only know how to break the generator assuming that the output length is a sufficiently large constant multiple than the seed length. It is not clear whether there is a linear test, or even a polynomial time algorithm, that breaks the case $k = 3$ when, say, $m = n + 1$.

It is still open whether there can be an ε-biased generator with negligible ε in the case $k = 4$. We conjecture that this is not the case for sufficiently large linear stretch, but we do not have a strong feeling about what happens for very small stretch.

The main open question is whether our generator for the case $k = 5$ can be broken by a polynomial time algorithm and, in general, whether polynomial time algorithms can break all NC^0 generators.

Another important open problem which may be more accesible it to understand the right asymptotics for ε-biased generators for large k. It is tempting to conjecture that either the upper bound $n^{O(k)}$ or the lower bound $n^{\Omega(\sqrt{k})}$ are actually tight.

Acknowledgements

We wish to thank David Wagner suggesting the relevance of correlation attacks. A.S. would also like to thank Avi Wigderson for helpful discussions.

References

[AC00] Noga Alon, Michael Capalbo. Explicit Unique-Neighbor Expanders. Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science, pages 73-79, 2000.

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.

[BKPS98] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. On the complexity of unsatisfiability proofs for random k-cnf formulas. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998.

[BOT02] Andrej Bogdanov, Kenji Obata, and Luca Trevisan. A lower bound for testing 3-colorability in bounded degree graphs. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 93–102, 2002.

[BSW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow: Resolution made simple. *Journal of the ACM*, 48(2), 2001.

[Cap01] Michael Capalbo. Explicit Constant-Degree Unique-Neighbor Expanders, 2001.

[CM01] Mary Cryan and Peter B. Miltersen. On pseudorandom generators in NC0. In *Proceedings of MFCS'01*, 2001.

[CRVW00] Michael Capalbo, Omer Reingold, Salil Vadhan and Avi Wigderson. Randomness Conductors and Constant-Degree Expansion Beyond the Degree/2 Barrier. Proceedings of the 34th Symposium on the Theory of Computing, 659-668, 2000.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. Technical Report TR00-090, ECCC, 2000.

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. Preliminary version in *Proc. of STOC'94*.

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 1–10, 1997.

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[Hir79] J. W. P. Hirschfeld, Projective Geometries over Finite Fields, Oxford University Press, 1979.

[IN96] R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of Cryptology*, 9(4):199–216, 1996.

[JJ99] T. Johansson and F. Jonsson. Improved fast correlation attacks on stream ciphers via convolutional codes. In *Proceedings of EUROCRYPT'99*, 1999.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[LRVW03] Chi-Jen Lu and Omer Reingold and Salil Vadhan and Avi Wigderson Extractors: Optimal Up to Constant Factors. To appear in proceedings of the 35th Annual symposium on the theory of computing (STOC).

[LR88] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 2(17):373–386, 1988.

[Mes95] . R. Meshulam. Spaces of Hankel matrices over finite fields, *Linear Algebra Appl.* **218**, 73–76, 1995.

[MOS03] E. Mossel, R. O'Donnell and R. Servedio (2003) Learning Juntas. To appear in proceedings of the 35th Annual symposium on the theory of computing (STOC).

[NN93] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications, 1993.

[Roth91] Ron Roth. Maximum rank array codes and their application to crisscross error correction, *IEEE Trans. on Info. Th.* **37,** 328–336, 1991.

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

[Shp02] A. Shpilka. On the rigidity of matrices. Manuscript, 2002.

[Vaz86] U. Vazirani. *Randomness, Adversaries and Computation.* PhD thesis, University of California, Berkeley, 1986.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and algebraic computation (EUROSAM '79, Internat. Sympos., Marseille, 1979)*, pages 216–226. Springer, Berlin, 1979.

A A Fourier lemma

In this section we prove Lemma 2 via the Fourier representation of the function. For a boolean function $f : \{0,1\}^k \to \{0,1\}$ we write F for the function $F : \{\pm 1\}^k \to \{\pm 1\}$ defined as

$$F((-1)^{x_1}, \ldots, (-1)^{x_k}) = (-1)^{f(x_1,\ldots,x_k)}$$

For a set $S \subset [k]$, we let $u_S : \{\pm 1\}^k \to \{\pm 1\}$ be defined as $u_S(X) = \prod_{i \in S} X_i$. It is well known that $\{u_S\}_{S \subset [k]}$ is an orthonormal basis. We write $F(x) = \sum_S \hat{F}(S) u_S(x)$ for the representation of F in the basis u_S.

We now prove Lemma 2.

Proof:

- Let $f : \{0,1\}^k \to \{0,1\}$ be a non-affine function. We prove that there exists a set S of size at most $k-2$ such that $\hat{F}(S) \neq 0$. This implies that F is correlated with u_S and therefore that f is correlated with $\oplus_{i \in S} x_i$ as needed.

  Look at the function $f \oplus \oplus_{i=1}^k x_i$. Since f is non-affine, this function is not a constant function.

  This function has the Fourier representation

  $$u_{[k]} F = \sum_S \hat{F}([k] \setminus S) u_S.$$

  It therefore suffices to prove that $u_{[k]} F$ has a coefficient $\hat{F}(S) \neq 0$ with $|S| \geq 2$. We will prove that any function which depends on more than one bit, has a non-zero coefficient with $|S| \geq 2$.

  Indeed, assume the contradiction

  $$F = a_0 + \sum_i a_i U_{\{i\}}$$

  For a $\pm$ vector X, write X^i for the vector where the i'th coordinate of X is multiplied by -1. Note that for all i and all X, it holds that $F(X) - F(X^i) \in \{0, \pm 2\}$, which implies that $a_i \in \{0, \pm 1\}$. Parsavel identity implies that $\sum a_i^2 = 1$. We therefore conclude that $F(X)$ depends on one bit as needed.

- Note that f is correlated with $\oplus_{i \in S} x_i$ if and only if $\hat{F}(S) \neq 0$. Moreover,

  $$\mathbf{Pr}[f(x) = \oplus_{i \in S} x_i] = \frac{1 + \hat{F}(S)}{2}.$$

  The claim will therefore follow once we prove that if

  $$F = \sum_{|S| \geq d} \hat{F}(S) u_S,$$

  and $\hat{F}(S) \neq 0$ for a set S of size d, then $|\hat{F}(S)| \geq 2^{d+1-k}$.

By looking at $u_{[k]}F$ instead of F, it suffices to prove that if

$$F = \sum_{|S| \leq k-d} \hat{F}(S) u_S, \tag{5}$$

and S' is a set of size $k-d$ such that $\hat{F}(S') \neq 0$, then $|\hat{F}(S')| \geq 2^{d-k+1}$. In order to prove the above claim, fix an X and look at the expression

$$A(X) = \sum_{T \subset S'} (-1)^{|T|} f(X^T),$$

where X^T is X where the coordinates at T are flipped (multiplied by -1). It is then clear that A obtains an integer value in the interval $2 \times [-2^{k-d-1}, 2^{k-d-1}]$.

On the other hand, since for every set S which doesn't contain S' and for all X, it holds that

$$\sum_{T \subset S'} (-1)^{|T|} u_S(X^T) = 0.$$

It follows that for all X

$$A(X) = 2^{k-d} \hat{F}(S') u_S(X).$$

We therefore conclude that $|\hat{F}(S')|$ obtain its values in $2^{-d+k+1} \times [-2^{k-d-1}, 2^{k-d-1}]$. In particular, since $\hat{F}(S') \neq 0$, it follows that $|\hat{F}(S')| \geq 2^{-d+k+1}$ as needed.

$\square$

B Small tests via expansion

In this section we prove lemmas 13 and 23.

Let $G = (L, R, E)$ be a bi-partite graph. G has the b - *right unique neighbor* property, if for any set $V \subset L$, $|V| \leq b$ we have that there exist a vertex $u \in R$ such that $|N(u) \cap V| = 1$.

Assign the n input variables to the different vertices in R. For every vertex $v \in L$ the corresponding output is the linear function

$$\ell_v(X) = \sum_{i \in N(v)} x_i$$

Lemma 33 *If G has the b-right unique neighbor property then any linear combination, $\ell = \sum_{v \in B} \ell_v$, of $b = |B|$ outputs is linearly independent.*

Proof: We have that

$$\ell = \sum_{v \in B} \ell_v = \sum_{i : |N(i) \cap B| = \text{odd}} x_i$$

The right unique neighbor property guarantees that there is an input variable that belongs to exactly one output. Therefore ℓ is not zero. $\square$

Note that we actually need the odd-neighbor property, but our calculations show that the graphs that we use have the stronger unique-neighbor property. The problem of constructing explicit expanders with the unique neighbor property was extensively studied in recent years and many new constructions were found [AC00, Cap01, CRVW00, LRVW03].

However, none of these construct the graph that we need. Thus we only prove the existence of such a graph instead of giving an explicit construction. Our proof actually show that if we pick a random graph (with the correct parameters) then w.h.p. it will have the unique-neighbor property.

The existence of graphs with the unique neighbour property will follow from the existence of certain expanders. We say that a bipartite graph (L, R, E) is (σ, α)-expanding if for every subset $S \subseteq L$ of vertices on the left, if $|S| \leq \sigma$ then $|\Gamma(S)| > \alpha \cdot |S|$, where $\Gamma(S)$, defined as

$$\Gamma(S) = \{v \in R : \exists u \in S \text{ such that } (u, v) \in E\}$$

is the neighborhood of S.

Lemma 34 *If $|\Gamma(S)| > \Delta|S|/2$ for all sets $S \subseteq L$ of size at most σ, then G has the σ-right unique neighbour property.*

PROOF: If there is no unique neighbour, then by counting edges $|\Gamma(S)| \leq \Delta|S|/2$. □

We need the following two lemmas

Lemma 35 *For every $c(n) = o(\sqrt{n}/(\log n)^{3/4})$ and sufficiently large n there is a $(\sigma, 3/2)$-expanding graph $([c(n) \cdot n], [n], E)$ with $\sigma = n/(4e^4c^2(n))$ such that every vertex on the left has degree 3.*

Note that Lemma 35 implies Lemma 13 via lemmas 34 and 33.

Proof of Lemma 35: We construct the graph at random by connecting each vertex on the left to three distinct randomly chosen vertices on the right. (For different left vertices the random choices are independent.)

Fix a size s, $3 \leq s \leq n/(2e^2c)$, and consider the probability that there is a subset $S \subseteq [cn]$ of s vertices on the right whose neighborhood is contained into a set $T \subseteq [n]$ of $3s/2$ vertices on the left. This probability is less than $(\frac{3s}{2n})^{3s}$. The number of possible choices for S is $\binom{cn}{s}$ and the number of possible choices for T is $\binom{n}{3s/2}$, and, by a union bound, the probability that the construction fails to satisfy the required property is at most

$$\sum_{s=3}^{\sigma} \binom{cn}{s} \cdot \binom{n}{3s/2} \left(\frac{3s}{2n}\right)^{3s} \tag{6}$$

and using the inequality $\binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$ we can see that Expression (6) is at most

$$\begin{aligned} & \sum_{s=3}^{\sigma} \left(\frac{ecn}{s}\right)^s \cdot \left(\frac{2en}{3s}\right)^{3s/2} \cdot \left(\frac{3s}{2n}\right)^{3s} && (7) \\ \leq \; & \sum_{s=3}^{\sigma} \left(\frac{2e^2c\sqrt{s}}{\sqrt{n}}\right)^s && (8) \\ = \; & O\left(\left(\frac{c}{\sqrt{n}}\right)^3 + \left(\frac{c}{\sqrt{n}}\right)^4 \cdot (\log n)^3\right) = o(1) && (9) \end{aligned}$$

Where the last line can be verifier by breaking the sum in Expression (8) up into the the term $s = 3$, which is $O((c/\sqrt{n})^3)$, the terms $s = 4, \ldots, 2\log n$, each of which is at most $O(c\sqrt{\log n}/\sqrt{n})^4$, and the remaining terms, each of which is at most $1/n^2$. $\square$

Similarly we can prove.

Lemma 36 *There exists a family of bi-partite graph $G_n = (L_n, R_n, E_n)$ with $|L| = n^t$, $|R| = n$, $\forall v \in L\ \ \deg(\mathrm{v}) = \Delta$, such that G_n has the n^ε-right unique neighbor property.*

One should think about the parameters in the following way $t \approx \sqrt{k}$, $\Delta = 10t$, $\varepsilon = \frac{1}{2}$.

Proof of Lemma 36: Let $|R| = n$, $|L| = n^t$. Connect every vertex in L to a randomly chosen multi set of size Δ of right vertices (that is we allow multiple edges between two vertices). As in lemma 35 we get that w.h.p. any set S, such that $|S| \le n^\varepsilon$, has at least $\frac{2\Delta}{3}|S|$ neighbors. Lemma 34 implies that S has a unique neighbor. $\square$

Combining lemmas 33, 36 we get Lemma 23.