On pseudorandom generators in $\mathbf{NC^0}$[⋆]

Mary Cryan and Peter Bro Miltersen

BRICS, Basic Research in Computer Science, Centre of the Danish National Research Foundation, Department of Computer Science, University of Aarhus.
{maryc,bromille}@brics.dk.

Abstract. In this paper we consider the question of whether $\mathbf{NC^0}$ circuits can generate pseudorandom distributions. While we leave the general question unanswered, we show

- Generators computed by $\mathbf{NC^0}$ circuits where each output bit depends on at most 3 input bits (i.e, $\mathbf{NC^0_3}$ circuits) and with stretch factor greater than 4 are not pseudorandom.
- A large class of "non-problematic" $\mathbf{NC^0}$ generators with superlinear stretch (including all $\mathbf{NC^0_3}$ generators with superlinear stretch) are broken by a statistical test based on a linear dependency test combined with a pairwise independence test.
- There is an $\mathbf{NC^0_4}$ generator with a super-linear stretch that passes the linear dependency test as well as k-wise independence tests, for any constant k.

1 Introduction

The notion of deterministically expanding a short seed into a long string that looks random to efficient observers, i.e., the notion of a *pseudorandom generator*, has been a fundamental idea in complexity as well as cryptography. Nevertheless, the question of whether strong pseudorandom generators actually exist is a huge open problem, as their existence (at least with cryptographic parameters) implies that $\mathbf{P} \neq \mathbf{NP}$: One can prove that a generator G is not pseudorandom by presenting a polynomial-time algorithm to decide range(G). Therefore if $\mathbf{P} = \mathbf{NP}$, strong pseudorandom generators do not exist.

On the other hand, that G has an $\mathbf{NP}$-hard range does by no means guarantee or even suggest that G is a strong generator, as there may be statistical regularities in the output of the generator. For this reason, existing pseudorandom generators are proven to be strong under stronger hardness assumptions than $\mathbf{P} \neq \mathbf{NP}$. Some of the most important research in this area was presented in a series of papers by Blum and Micali [2], Yao [18] and Goldreich and Levin [6]. Håstad, Impagliazzo, Levin and Luby [9, 8], building on this research, finally showed that the existence of a cryptographic one-way function is sufficient (and trivially necessary) for the existence of a cryptographic pseudorandom generator. This general construction only depends on a generic one-way function, but

[⋆] Full version at http://www.brics.dk/∼maryc/publ/psu.ps. Partially supported by the IST Programme of the EU, contract number IST-1999-14186 (ALCOM-FT).

the resulting generator is quite involved and seems to need the full power of the complexity class $\mathbf{P}$.

The general construction left open the precise computational power needed to produce pseudorandomness. Kharitonov [11] showed, under a more specific hardness assumption, that there is a secure pseudorandom generator in $\mathbf{NC^1}$ for any polynomial stretch function. Impagliazzo and Naor [10] showed how to construct secure pseudorandom generators based on the assumed intractability of the subset sum problem. In particular, they showed how to construct a generator with a non-trivial stretch function (expanding n bits to $n + \Theta(\log(n))$ bits) in the complexity class $\mathbf{AC}^0$. This suggests that even rudimentary computational resources are sufficient for producing pseudorandomness (it is worth noting that Linial, Mansour and Nisan [14] proved that there are no pseudorandom *function* generators in $\mathbf{AC^0}$ with very good security parameters).

From both a theoretical and a practical point of view it seems worth finding out *how* rudimentary pseudorandom generators can be. There are different interpretations of this question, depending on how we formalise "rudimentary" or "simple". In previous work Kharitonov et al. [12] and Yu and Yung [19] proved strong negative results about the ability of various automata and other severely space-restricted devices to produce pseudorandomness. In this paper, we interpret "simple" in terms of circuit complexity and ask: *Are there pseudorandom generators in* $\mathbf{NC^0}$?, that is, are there pseudorandom generators where each output bit depends on only a constant number of input bits? Such generators would be appealing in practice as they can be evaluated in constant time using only bounded fan-in hardware. However, it seems that with such a severe constraint on the generator, there would certainly be statistical regularities in the output, so that we could construct a sequence of circuits to distinguish between the uniform distribution and output from the generator. This is a tempting conjecture, but we have not been able to prove it without further restrictions.

Our main results are:

Theorem 3 *There is no strong pseudorandom generator with a stretch factor greater than* 4 *for which every output bit depends on at most* 3 *input bits (that is, no* $\mathbf{NC^0_3}$*-generator).*

We can actually prove a weaker version of the theorem above for a more general class of "non-problematic" generators of which $\mathbf{NC^0_3}$ generators are a special case, and we do this in Theorem 1. It is interesting to note that deciding the *range* of a $\mathbf{NC^0_3}$ circuit is, in general, $\mathbf{NP}$-hard (see Proposition 1), so we cannot use the approach of inverting the generator's output to prove either of these two theorems. Instead, in Theorem 1 we show that we can break any such generator with a statistical test consisting of a linear dependency test and a test for pairwise independence of the output bits. On the other hand, we show

Theorem 5 *There is a generator with a superlinear stretch for which every output bit depends on at most 4 input bits (i.e., an* $\mathbf{NC^0_4}$*-generator) so that that there are no linear dependencies among the output bits and so that for any* *constant k and for sufficiently large input size, all bits of the output are k-wise independent.*

The question of whether there are true pseudorandom generators in $\mathbf{NC^0_4}$ is still open. We have no construction that we believe is truly pseudorandom, nor do we have a general scheme for breaking such generators. We have been able to reduce the number of boolean functions on 4 variables that could possibly serve as basis for such a generator to 4, up to isomorphism.

2 Definitions and Background

Definition 1. A circuit $C : \{0,1\}^n \to \{0,1\}^N$ is in $\mathbf{NC^0_c}$ if every output bit of C is a function of at most c input bits. The circuit C is associated with an induced distribution on $\{0,1\}^N$, where the probability of $y \in \{0,1\}^N$ is the probability that C outputs y when the input to C is chosen from the uniform distribution U_n on $\{0,1\}^n$. We will be flexible about notation and use C to denote this induced distribution as well as the circuit itself. We say a *sequence* of circuits $\{C_n\}_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is in $\mathbf{NC^0_c}$ if C_n is in $\mathbf{NC^0_c}$ for all n. The function $\ell(n)$ is known as the *stretch function* of the sequence and the *stretch factor* is $\ell(n)/n$.

Definition 2. A function $f : \mathbf{N} \to [0,1]$ is said to be *negligible* if for every $c > 0$, there is some constant n_c such that $f(n) < n^{-c}$ for every $n \geq n_c$. It is said to be *overwhelming* if $1 - f$ is negligible.

Definition 3 (from [10]). A function $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is a *pseudorandom generator* if every non-uniform polynomial-time algorithm A has a *negligible* probability of distinguishing between outputs of G and truly random sequences; that is, for every algorithm A,

$$|\Pr[A(G(x)) = 1] - \Pr[A(y) = 1]| \quad \text{is negligible,}$$

where x and y are chosen from U_n and $U_\ell(n)$ respectively.

Deviating slightly from the above definition, we prove, for convenience, our non-pseudorandomness results for sequences of $\mathbf{NC^0_c}$ distributions by using a *statistical test* to distinguish between *several* samples of the uniform distribution and several samples of the $\mathbf{NC^0_c}$ distributions, rather than one sample.

Definition 4. An efficient *statistical test* is a procedure which takes as input a parameter N and $m = m(N)$ *samples* (for some $m(N) = N^{O(1)}$) $y_1, \ldots, y_m$ from $\{0,1\}^N$, runs in polynomial time, and either ACCEPTs or REJECTs. Furthermore, if $y_1, \ldots, y_m$ are chosen from the uniform distribution U_N, the probability of acceptance is overwhelming (that is, the probability of rejection is negligible).

Definition 5. Let A be a statistical test. An ensemble $\{D_n\}_{n\in\mathbf{N}}$ of probability distributions on $\{0,1\}^{\ell(n)}$, $\ell : \mathbf{N} \to \mathbf{N}$ is said to *fail* A if the probability that A rejects $y_1, \ldots, y_{m(\ell(n))}$ chosen independently from D_n is *not* negligible. Otherwise the ensemble is said to *pass* A.

Although Definition 3 is the traditional definition of pseudorandomness, it is well known that an ensemble of distributions generated by a generator fails *some* (possibly non-uniform) statistical test in the sense of Definition 5 if and only if the generator is *not* pseudorandom in the sense of Definition 3 (see Goldreich [5, page 81] for details).

Some of the results in Sections 3 and 4 refer to special $\mathbf{NC^0_c}$ circuits that we refer to as *non-problematic* circuits. We need the following definition:

Definition 6. A function $f : \{0,1\}^c \to \{0,1\}$ is *affine* if $f(x_1, \ldots, x_c)$ can be written in the form $\alpha_1 x_1 + \ldots \alpha_c x_c + \alpha_{c+1} \pmod 2$, for $\alpha_i \in \{0,1\}$. We say that f is *statistically dependent* on the input variable x_i if either $\Pr[f(x) = 1 \mid x_i = 1] \neq 1/2$ or $\Pr[f(x) = 1 \mid x_i = 0] \neq 1/2$. An $\mathbf{NC^0_c}$ circuit is *non-problematic* if for every output gate $y_j = f_j(x_{j,1}, \ldots, x_{j,c})$, the function f_j is either an affine function or depends statistically on one of its variables.

Before we present our results, we discuss previous negative results on generation of pseudorandomness. Lower bounds for pseudorandom *function* generators were studied in [14, 16, 13]. The only papers we are aware of that give impossibility results or lower bounds for "plain" pseudorandom generators as defined above, are the papers by Kharitonov et al. [12] and Yu and Yung [19].

Kharitonov et al. [12] proved that no generator that is either a one-way logspace machine or a one-way pushdown machine is a strong generator, even when the generator is only required to extend the input by a single bit. They also considered the question of whether sublinear-space generators exist and related this issue to the $\mathbf{L}$ vs $\mathbf{P}$ question. It is worth noting that the proofs of non-pseudorandomness that Kharitonov et al. obtain for one-way logspace machines and one-way pushdown machines (and for finite reversal logspace machines) depend on showing that the range of these generators can be recognized in polynomial time.

Yu and Yung [19] considered bidirectional finite state automata and bidirectional machines with sublogarithmic space ($o(\log n)$ space) as generators. They proved that bidirectional finite state automata are not strong pseudorandom generators, even when the generator only extends the input by one bit. For these generators, they also showed that the tester for distinguishing between the generator's output and the uniform distribution can be assumed to lie in $\mathbf{L}^2$. For bidirectional machines with $o(\log n)$-space, they showed that any generator in this class with superlinear stretch is not pseudorandom. All the impossibility results of Yu and Yung are again obtained by showing that the range of the generator can be recognized in polynomial time.

We can use a similar strategy to show that pseudorandom generators in $\mathbf{NC^0_2}$ do not exist: Note that for all uniform sequences $\{C_n\}_n$ of circuits in $\mathbf{NC^0_2}$, range(C_n) is in $\mathbf{P}$. This follows because the problem is a special case of 2-SAT. However, this is a strategy we cannot use (in general) for distributions generated by $\mathbf{NC^0}$-circuits, because we now show that there are sequences $\{C_n\}_n$ of circuits in $\mathbf{NC^0_3}$ such that deciding range(C_n) is $\mathbf{NP}$-complete. The proposition below is an improvement of a result due to Agrawal et al. [1, Proposition 1], who showed that there is a function f in $\mathbf{NC^0_4}$ such that that is **NP**-complete to invert f. It also improves an unpublished theorem of Valiant (Garey and Johnson [4], page 251), stating that inverting multinomials over $GF(2)$ is **NP**-hard: we prove that this is the case even for a specific sequence of multinomials.

Proposition 1. *There is a uniform sequence* $\{C_n\}_{n\in\mathbf{N}}$ *of* $\mathbf{NC^0_3}$ *circuits such that* $\mathrm{range}(C_n)$ *is* **NP***-complete. Also, every output gate in every circuit in the sequence is a degree-*2 *multinomial over* $GF(2)$.

Proof. The reduction is from the **NP**-complete problem 3-SAT . For every $n \in \mathbf{N}$ we construct a $\mathbf{NC^0_3}$ circuit to model 3-SAT problem on n variables. The circuit has an input gate x_i for $1 \le i \le n$ for each logical variable, and three helper inputs $h_{j,1}, h_{j,2}, h_{j,3}$ for every possible three-literal clause c_j (there are $8\binom{n}{3}$ c_js). The circuit has four output gates $y_{j,0}$, $y_{j,1}, y_{j,2}, y_{j,3}$ for every c_j. Let $N = \frac{16}{3}n(n-1)(n-2)$ denote the total number of output gates.

The gates are connected to form a $\mathbf{NC^0_3}$ circuit as follows: for every possible clause c_j, we connect $h_{j,1}, h_{j,2}$ and $h_{j,3}$ (the "helper" inputs) to $y_{j,0}$ as follows:

$$y_{j,0} =_{def} (1 + h_{j,1} + h_{j,2} + h_{j,3}) \pmod 2 \tag{1}$$

It is easy to check that $y_{j,0} = 1$ iff an even number (0 or 2) of the helper inputs $\{h_{j,1}, h_{j,2}, h_{j,3}\}$ are switched on. Let the literals in clause c_j be $\ell_{j,1}$, $\ell_{j,2}$ and $\ell_{j,3}$. Each of the output gates $\{y_{j,1}, y_{j,2}, y_{j,3}\}$ is the disjunction of one helper input and one of the literals in c_j:

$$y_{j,i} =_{def} 1 - (1 - \ell_{j,i})(1 - h_{j,i}) \pmod 2 \quad i = 1, 2, 3 \tag{2}$$

Two facts:

(i) If all of the helper inputs for c_j are 1 then $y_{j,0} = 0$ and $y_{j,1}y_{j,2}y_{j,3} = 111$ both hold.

(ii) Suppose the x_i inputs are fixed but the helper inputs are not. Then we can set values for $h_{j,1}$, $h_{j,2}, h_{j,3}$ to give $y_{j,0}y_{j,1}y_{j,2}y_{j,3} = 1111$ iff at least one of the literals in c_j is true under the truth assignment given by the x_i input variables.

For any instance (X, C) of 3-SAT with $n = |X|$ variables, define $a \in \{0,1\}^N$ by setting $a_{j,0}a_{j,1}a_{j,2}a_{j,3} = 1111$ for every $c_j \in C$ and $a_{j,0}a_{j,1}a_{j,2}a_{j,3} = 0111$ for every $c_j \notin C$. By fact (i), we can imagine that all the helper inputs for clauses outside C are set to 1. Then by fact (ii), $a \in range(C_n)$ iff there is some truth assignment for the x_i variables that satisfies some literal of every clause in C.

Note that the functions defined for the output gates (Equations 1 and 2) are all degree 2 multinomials.

3 Lower Bounds

In this section we prove that sequences of non-problematic $\mathbf{NC^0_c}$ circuits with a "large enough" (constant) stretch factor are not pseudorandom generators.

The particular statistical test that we use to detect non-pseudorandomness is the LINPAIR test, which tests for non-trivial linear dependencies in the output (Step (i)) and also tests that the distribution on every pair of indices is close to uniform. We also use a generalization of the LINPAIR test called the LIN(k) test, which instead of merely checking that the distribution on every *pair* of outputs is close to uniform, checks that the distribution on every group of k indices is close to uniform (Step (ii)). Note that Step (i) of LIN(k) can be computed in polynomial-time by Gaussian elimination. Also for any constant $k \in \mathbf{N}$, Step (ii) of LIN(k)$[m, N]$ is polynomial in m and N, because there are only $2^k \binom{N}{k}$ different $F_{j_1,\ldots,j_k}[b_1, \ldots, b_k]$ values to calculate and test.

Throughout the paper we will use LINPAIR$[m, N]$ to denote LIN(2)$[m, N]$.

Algorithm 1 LIN(k)$[m, N]$

```
input: $m$ samples $a_1, \ldots, a_m \in \{0,1\}^N$. Each $a_i$ is written as $a_{i,1} \ldots a_{i,N}$.
output: ACCEPT or REJECT

(i) Check the linear system $\sum_j z_j a_{i,j} = z_{N+1} (\text{mod } 2)$ $(1 \le i \le m)$ for a
non-trivial solution;
If there is a non-trivial solution then REJECT;
else
   (ii) For every set of $k$ indices $j_1, \ldots, j_k$ $(1 \le j_1 < \ldots j_k \le N)$, calculate
        $F_{j_1,\ldots,j_k}[b_1, \ldots, b_k] =_{def} \frac{1}{m}|1 \le i \le m : a_{i,j_1} \ldots a_{i,j_k} = b_1 \ldots b_k]$,
        for every tuple $b_1 \ldots b_k \in \{0,1\}^k$;
   if $|F_{j_1,\ldots,j_k}[b_1, \ldots, b_k] - (1/2^k)| > \frac{1}{N}$ for any $j_1, \ldots, j_k$ and $b_1, \ldots, b_k$
   then
        REJECT;
   else
        ACCEPT;
   end
end
```

Now we show that LIN(k) is a statistical test for any k (it accepts the uniform distribution):

Lemma 1. *Let $m : \mathbf{N} \to \mathbf{N}$ be any function satisfying $m(N) \ge N^2 \log^2 N$. Then if the* LIN(k) *algorithm is run with m samples from U_N, the probability that* LIN(k) *accepts is overwhelming.*

Proof. First consider Step (i). First note that since we are working with binary arithmetic modulo 2, any non-trivial solution to the linear system $\sum_{j=1}^{N} z_j a_{i,j} = z_{N+1} (\text{mod } 2)$ corresponds to a non-empty subset $S \subseteq \{1, \ldots, N\}$ such that either (a) $\sum_{j \in S} a_{i,j} = 0$ holds for all $1 \le i \le m$ or (b) $\sum_{j \in S} a_{i,j} = 1$ holds for all i. For any fixed set S of indices, the probability that $\sum_{j \in S} a_{i,j} = 0$ holds for a single a_i chosen from the uniform distribution is $1/2$. The probability that (a) holds for all $1 \le i \le m$ is $1/2^m$, and the probability that either (a) holds or (b) holds is $1/2^{m-1}$. There are only 2^N subsets of $\{1, \dots, N\}$, so we can bound the probability of finding any non-trivial solution to $\sum_{j=1}^{N} z_j a_{i,j} = z_{N+1} (\text{mod } 2)$ by $2^N/2^{m-1}$, which is at most 2^{-N} when $m \geq N^2 \log^2 N$.

Next we bound the probability that Step (ii) rejects. For any k indices $j_1, \dots, j_k$, and any k-tuple of bits $b_1, \dots b_k \in \{0,1\}^k$, the probability that $j_1, \dots, j_k$ reads $b_1 \dots b_k$ in a sample from U_N is $1/2^k$. By Hoeffding's Inequality (see McDiarmid [15, Corollaries (5.5) and (5.6)]), if we take m samples from U_N, then

$$\begin{aligned}\Pr\left[|F_{j_1,\dots,j_k}[b_1,\dots,b_k] - 1/2^k| \geq 1/N\right] &\leq 2\exp[-1/3(2^k/N)^2 m(1/2^k)] \\ &\leq 2\exp[-2^k/3(\log N)^2]\end{aligned}$$

which is at most $2N^{-(\log N)2^{k-2}}$. There are $2^k N^k$ different tests in Step (ii), so the probability that any one of these fails is at most $2^{k+1}N^{k-(\log N)2^{k-2}}$, which is at most $N^{-\log N/2}$ for large N.

So LIN(k) succeeds with probability at least $1 - 2^{-N} - N^{-\log N/2}$.

Theorem 1. *For any constant $c \geq 2$ there is a constant $d \in \mathbf{N}$ so that the following holds: If $\{C_n\}_{n\in\mathbf{N}}$ is a family of non-problematic $\mathbf{NC^0_c}$ circuits whose stretch function satisfies $\ell(n) \geq dn$, then $\{C_n\}_{n\in\mathbf{N}}$ fails* LINPAIR$[m, \ell(n)]$ *for any function m with $m(\ell(n)) \geq \log^2 \ell(n)$. Hence $\{C_n\}_{n\in\mathbf{N}}$ is not pseudorandom.*

Proof. The constant $d = (2^{(c-1)2^c}(c-1)!)+1$ is large enough for this theorem. We will show that for all such circuits, LINPAIR$[m, \ell(n)]$ fails for all $m \geq \log^2 \ell(n)$ (for large enough n).

There are two cases. The circuits are non-problematic, so every output bit of C_n is either an affine bit or is statistically dependent on one of its input variables.

Case (i): First we prove the theorem for the case when C_n has at least $n+1$ affine outputs. Without loss of generality, let the affine outputs be $y_1, \dots, y_{n+1}$. Since there are only n input variables, and each output y_i is an affine combination of some input variables, there is a non-trivial affine dependence among the y_i's. That is, we can find constants $\alpha_1, \dots, \alpha_{n+1} \in \{0,1\}$ such that not all of the α_i are 0 and such that either $\sum_{i=1}^{n} \alpha_i y_i = 0 (\text{mod } 2)$ or $\sum_{i=1}^{n} \alpha_i y_i = 1 (\text{mod } 2)$ holds for all $x_1, \dots, x_n \in \{0,1\}$. Therefore, Step (i) of the LINPAIR algorithm will always find a non-trivial solution (regardless of the number m of samples), and the algorithm will reject.

Case (ii): If there are less than $n+1$ affine output bits, then there are at least $n(d-1)$ output bits that are statistically dependent on at least one of their input variables. Therefore there is at least one input bit x such that $(d-1)$ different output bits are all statistically dependent on x. Assume wlog that these output bits are $y_1, \dots, y_{d-1}$. We will prove that the distribution on some pair of these bits deviates from the uniform distribution U_2, and therefore will fail Step (ii) of LINPAIR.

We use the Erdős-Rado sunflower lemma (see Papadimitriou[17, page 345]), which gives conditions that are sufficient to ensure that a family $\mathcal{F}$ of sets contains a sunflower of size p, where a sunflower is a subfamily $\mathcal{F}' \subset \mathcal{F}$ of size p, where every pair of subsets in $\mathcal{F}'$ has the same intersection. The family $\mathcal{F}$ that we consider contains a set I_i for every $1 \le i \le d-1$, where I_i contains all the input variables feeding y_i except x. By the sunflower lemma, our definition of d ensures that there is a sunflower of size $d' = 2^{2^c} + 1$ in $\mathcal{F}$. That is, we have d' special output bits $y_1, \dots, y_{d'}$, such that each y_i depends on the bit x, on some additional input bits $u = (u_1, \dots, u_r)$ common to $y_1, \dots, y_{d'}$, and on some extra input bits z_i not shared by any two outputs in $y_1, \dots, y_{d'}$.

There are only 2^{2^c} different functions on c variables, so if we take the most common function f among the output bits $y_1, \dots, y_{d'}$ (with the input bits ordered as x first, then u, then the private input bits), then we can find at least 2 output bits with the same function. Assume these are y_1 and y_2. So we have

$$y_1 = f(x, u_1, \dots, u_r, z_1) \text{ and } y_2 = f(x, u_1, \dots, u_r, z_2)$$

where z_1 and z_2 are vectors of input bits, and z_1 and z_2 are disjoint and of the same length. We can now show that the distribution on (y_1, y_2) is not U_2. Calculating, we have

$$\begin{aligned}
\Pr[y_1y_2 = 11] &= \frac{1}{2}(\Pr[y_1y_2 = 11|x = 1] + \Pr[y_1y_2 = 11|x = 0]) \\
&= \frac{1}{2}(\mathrm{Avg}_j(\Pr[y_1y_2 = 11|xu = 1j]) + \mathrm{Avg}_j(\Pr[y_1y_2 = 11|xu = 0j])) \\
&= \frac{1}{2}(\mathrm{Avg}_j(\Pr[y_1 = 1|xu = 1j]^2) + \mathrm{Avg}_j(\Pr[y_1 = 1|xu = 0j]^2)) \\
&= \frac{1}{2}(\mathrm{Avg}_j(\beta_j)^2 + \mathrm{Avg}_j(\gamma_j)^2)
\end{aligned}$$

where Avg_j denotes the average of its argument over all $j \in \{0,1\}^r$, and $\beta_j = \Pr[y_1 = 1|xu = 1j]$ and $\gamma_j = \Pr[y_1 = 1|xu = 0j]$. We know that the output y_1 is statistically dependent on x, so we assume wlog that $\Pr[y_1 = 1|x = 1] = 1/2 + \epsilon$, where $\epsilon \ge 1/2^c$ (since y_1 is a function on c inputs). Then, assuming $\Pr[y_1 = 1] = 1/2$, we have $\Pr[y_1 = 1|x = 0] = 1/2 - \epsilon$ (if we have $\Pr[y_1 = 1] \ne 1/2$, then we are already finished).

For any sequence of numbers $\{t_j\}$, $\mathrm{Avg}_j(t_j)^2 \ge (\mathrm{Avg}_j t_j)^2$. Also, $\Pr[y_1 = 1|x = 1] = \mathrm{Avg}_j \beta_j$ and $\Pr[y_1 = 1|x = 0] = \mathrm{Avg}_j \gamma_j$. Therefore

$$\begin{aligned}
\Pr[y_1y_2 = 11] &\ge \frac{1}{2}(\Pr[y_1 = 1|x = 1]^2 + \Pr[y_1 = 1|x = 0]^2) \\
&= \frac{1}{2}((1/2 + \epsilon)^2 + (1/2 - \epsilon)^2) \\
&= \frac{1}{2}(1/2 + 2\epsilon^2)
\end{aligned}$$

Now consider the calculation of $F_{1,2}[1,1]$ in Step (ii). The expected value of $F_{1,2}[1,1]$ is at least $1/4 + 1/2^{2c}$. Therefore, if the deviation of $F_{1,2}[1,1]$ from its expectation is bounded by $1/2^{2c+1}$, we have $|F_{1,2}[1,1] - 0.25| \ge 1/2^{2c+1}$. Using Chernoff Bounds (see McDiarmid [15]), the probability that $F_{1,2}[1,1]$ deviates from its expectation by $1/2^{2c+1}$ is at most $2\exp[-2m(\ell(n))/2^{2(2c+1)}]$. Thus for $m(\ell(n)) \ge \log^2 \ell(n)$ and large enough n, we have $|F_{1,2}[1,1] - 0.25| \ge 1/2^{2c+1} \ge$

$1/\ell(n)$ with probability at least $1 - 2\exp[-2(\log \ell(n))^{3/2}]$. Therefore for sufficiently large $\ell(n)$, Step (ii) fails and LINPAIR$[m, \ell(n)]$ rejects with probability at least $1 - 2\ell(n)^{-2(\log^{1/2}\ell(n))}$.

One of our referees has pointed out that the bound on d in Theorem 1 can be improved by testing the variance of sums of output variables. For any discrete random variable X, the variance of X, denoted $\mathrm{var}(X)$, is defined as the expectation of $(X - \mathbf{E}[X])^2$. In case (ii) of Theorem 1, we find a set of output variables $y_1, \ldots, y_{d-1}$ such that each y_i is statistically dependent on the same variable x. Suppose $d \geq 2^{2c} + 1$. Then $\mathrm{var}(\sum_{i=1}^{d-1} z_i)$ is at least $d-1$ for some choice of $z_i \in \{y_i, \bar{y}_i\} : i = 1, \ldots d-1$, whereas the variance is $(d-1)/4$ when the z_i are independent uniform random bits (see Grimmett and Stirzaker [7]). The following statistical test distinguishes between the $\mathbf{NC^0_c}$ circuits of case (ii) and the uniform distribution: for every $z_{i_1}, \ldots, z_{i_{d-1}}$ where $z_{i_j} \in \{y_i, \bar{y}_i\}$ and the i_j are distinct indices, calculate the average of $(\sum_{j=1}^{d-1} z_{i_j} - (d-1)/2)^2$ over the set of samples. If any of these estimates is greater than $(d-1)/2$, then reject.

The following observation can be easily verified by computer.

Observation 2 *Every $\mathbf{NC^0_3}$ circuit is non-problematic.*

By Observation 2, Theorem 1 holds for *all* sequences of $\mathbf{NC^0_3}$ circuits. By a more careful analysis of $\mathbf{NC^0_3}$ circuits, we can prove a stronger result:

Theorem 3. *Let $\{C_n\}_{n\in\mathbf{N}}$ be any ensemble of $\mathbf{NC^0_3}$ circuits which has a stretch function $\ell(n)$ with $\ell(n) \geq 4n+1$. Then $\{C_n\}_{n\in\mathbf{N}}$ fails* LIN(4)$[m, \ell(n)]$ *for any m satisfying $m(\ell(n)) \geq \log^2 \ell(n)$ and therefore is not pseudorandom.*

4 A generator that passes LIN(k)

Our original goal was to prove that for all $c \in \mathbf{N}$ there exists some constant $d \in \mathbf{N}$ such that any sequence of $\mathbf{NC^0_c}$ circuits with stretch factor at least d is not pseudorandom. So far we have only been able to prove this for sequences of non-problematic circuits. For the set of $\mathbf{NC^0_4}$ circuits, we can classify the number of different problematic output gate functions in the following way:

Observation 4 *Consider the set of all problematic functions $f : \{0,1\}^4 \to \{0,1\}$ on four variables. Define an equivalence relation on this set by saying $f \equiv g$ iff*

$$f(x_1, x_2, x_3, x_4) = b_5 + \ g(\pi(x_1 + \ b_1, \ \ldots, x_4 + \ b_4)) (\mathrm{mod}\ 2)$$

for some permutation π and some five boolean values $b_1, \ldots, b_5 \in \{0,1\}$. That is, two functions are equivalent if one can be obtained from the other by permuting and possibly negating input variables and possibly negating the output. Then, there are only four problematic non-equivalent functions on 4 inputs, namely

$$f_1(x_1, \ldots, x_4) = x_1 + x_2 + x_3 x_4 \ (\mathrm{mod}\ 2)$$

$$f_2(x_1, \ldots, x_4) = x_1 + x_2x_3 + x_3x_4 + x_2x_4 \pmod 2$$
$$f_3(x_1, \ldots, x_4) = x_1 + x_2 + x_4(x_2 + x_3) \pmod 2$$
$$f_4(x_1, \ldots, x_4) = x_1 + x_2 + (x_1 + x_4)(x_2 + x_3) \pmod 2$$

Proof. Case analysis (by computer).

Any $\mathbf{NC_4^0}$ circuit with a large enough stretch factor is guaranteed to either (a) contain enough non-problematic output gates to allow us to use Theorem 1 to prove non-pseudorandomness, or (b) contain a large number of output gates of the form f_i, for one of the $f_1, \ldots, f_4$ functions. So when we consider candidates for pseudorandom generators in $\mathbf{NC_4^0}$ with superlinear stretch, we only have to consider generators using gates of the form of *one* of f_1, f_2, f_3, f_4.

We show that Theorem 1 cannot be extended to non-problematic functions by exhibiting an $\mathbf{NC_4^0}$ generator using only output gates of the form f_1 and passing LIN(k) for all constants k. We need the following lemma (a variation of the well-known Schwartz-Zippel lemma):

Lemma 2. *For any r-variable multilinear polynomial f of degree at most 2 over $GF(2)$ which is not a constant function, $\Pr_x[f(x) = 0] \in [1/4, 3/4]$ when x is chosen uniformly at random from $\{0,1\}^r$.*

Theorem 5. *Let $k : \mathbf{N} \to \mathbf{N}$ be any function satisfying $2 \le k(n) < \log n$. Then there is a generator in $\mathbf{NC_4^0}$ with stretch function $\ell(n) = n^{1+\Theta(1/k)}$, such that for any function m satisfying $m \ge \ell(n)^2(\log^2 \ell(n))$, the generator passes* LIN$(k(n))[m,\ \ell(n)]$ *with overwhelming probability.*

Proof. In this proof we will informally use k to denote $k(n)$ and ℓ to denote $\ell(n)$. Our construction uses the following result from extremal graph theory (see Bollabás [3, page 104]): For all n there exists a graph G on $n/2$ nodes with ℓ edges, such that the girth of G (i.e., the length of the shortest cycle of G) is at least k. We use this graph to construct a circuit, with an output gate for every edge in G: the input bits of the generator are split into two sets, the set $\{z_1, \ldots, z_{n/2}\}$ which will represent the nodes of G, and another set $\{x_1, \ldots, x_{n/2}\}$. We assume some arbitrary enumeration $\{(u_i, v_i) :\ i = 1, \ldots, \ell\}$ of the edges of the graph, and we also choose any ℓ different ordered pairs $\{(i_1, i_2) :\ i = 1, \ldots, \ell\}$ from $\{1, \ldots, n/2\}^2$. Then, for every output bit y_i, we define

$$y_i =_{def} x_{i_1}x_{i_2} + z_{u_i} + z_{v_i} \pmod 2,$$

where z_{u_i} and z_{v_i} are the inputs from $\{z_1, \ldots, z_{n/2}\}$ representing u_i and v_i respectively. Note that $\Pr[y_i = 1] = 1/2$ for every i.

We show that the generator passes the linear dependency test of LIN(k) with high probability and also that the output bits are k-wise independent, and therefore pass Step (ii) of LIN(k).

First consider the test for linear dependence among the outputs $y_1, \ldots, y_\ell$ (Step (i)). By definition of the y_i functions, no two y_i share the same $x_{i_1}x_{i_2}$ term. Then for every sequence $\alpha_1, \ldots, \alpha_\ell \in \{0,1\}$ containing some non-zero term, $\sum_{i=1}^{\ell} \alpha_i y_i$ is not a constant function. Step (i) rejects (given m samples $a_1, \ldots, a_m \in \{0,1\}^\ell$) iff there exist $\alpha_1, \ldots, \alpha_\ell \in \{0,1\}$ not all zero such that

$$\sum_{i=1}^{\ell} \alpha_i a_{j,i} \quad \text{is constant for all } 1 \leq j \leq m$$

For any particular sequence $\alpha_1, \ldots, \alpha_\ell \in \{0,1\}$, Lemma 2 implies that the probability that m samples satisfy the equation above is at most $2(3/4)^m$. There are 2^ℓ different α-sequences, so the total probability that Step (i) rejects is at most $2^\ell 2(3/4)^m = 2^\ell 2(3/4)^{\ell^2 \log^2 \ell(n)} \leq (3/4)^{\ell^2/2}$, which is negligible.

For Step (ii), we show that any k outputs are mutually independent. Let k' be the minimum k' for which k' output bits are mutually dependent. Assume wlog that these output bits are $y_1, \ldots, y_{k'}$. Now, suppose there is some y_i with $1 \leq i \leq k'$ such that one of z_{u_i}, z_{v_i} does not appear in any other output function for $y_1, \ldots, y_{k'}$. If this is the case, then regardless of the values along the output gates $y_1 \ldots y_{i-1} y_{i+1} \ldots y_{k'}$, the probability that $y_i = 1$ is always $1/2$ (y_i is mutually independent of all the other y_i's). Therefore the set of outputs $y_1, \ldots y_{i-1}, y_{i+1} \ldots y_{k'}$ must be mutually dependent and we obtain a contradiction. Therefore if k' is the minimum value for which k' outputs are mutually dependent, then for every y_i, z_{u_i} and z_{v_i} appear in at least one other output from $y_1, \ldots, y_{k'}$. In terms of the original graph that we used to construct our circuit, we find that in the subgraph consisting of the set of edges for $y_1, \ldots, y_{k'}$, each vertex has degree at least 2. Then this subgraph contains a cycle of length at most k', so $k' \geq k$, as required.

Then an argument similar to the argument for LIN(k) in Lemma 1 shows that Step (ii) rejects with negligible probability.

5 Open problems

The main open problem is of course whether $\mathbf{NC^0}$ circuits in general can be pseudorandom generators. We believe this may turn out to be a difficult question. Some, perhaps, easier subquestions are the following:

We have been able to show that there is no pseudorandom generator in $\mathbf{NC^0_3}$ that expands n bits to $4n+1$ bits. It would be interesting to optimise this to show the non-existence of generators in $\mathbf{NC^0_3}$ expanding n bits to $n+1$ bits. . Another goal that may be within reach is to prove that $\mathbf{NC^0_4}$ generators with superlinear stretch cannot be pseudorandom generators. There is no reason to believe that the construction of Theorem 5 is unbreakable. Indeed, note that the generator is not specified completely as the graph and the exact enumeration of pairs is unspecified. It is easy to give examples of specific graphs and enumerations where the resulting generator can be easily broken by testing whether a particular linear combination of the output bits yields an unbiased random variable. If one believes that $\mathbf{NC^0_4}$ generators in general are breakable (as we tend to do), Observation 4 suggests that a limited number of *ad hoc* tests may be sufficient to deal with all cases. For instance, it is conceivable that every such generator is broken by testing whether a particular linear combination of the output bits yields an unbiased random variable.

References

1. M. Agrawal, E. Allender and S. Rudich, "Reductions in Circuit Complexity: An Isomorphism Theorem and a Gap Theorem"; *Journal of Computer and System Sciences*, Vol **57**(2): pages 127-143, 1998.
2. M. Blum and S. Micali, "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits"; *SIAM Journal on Computing*, Vol **13**: pages 850-864, 1984.
3. B. Bollabás, *Extremal Graph Theory*; Academic Press Inc (London), 1978.
4. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman and Company (1979).
5. O. Goldreich, *Modern Cryptography, Probabilistic Proofs and Pseudo-randomness*; Vol **17** of series on Algorithms and Combinatorics, Springer-Verlag, 1999.
6. O. Goldreich and L.A. Levin, "Hard-Core Predicates for any One-Way Function"; *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25-32, 1989.
7. G.R. Grimmett and D.R. Stirzaker, *Probability and Random Processes*, Oxford University Press, 1992.
8. J. Håstad, R. Impagliazzo, L.A. Levin and M. Luby, "A Pseudorandom Generator from any One-way Function"; *SIAM Journal on Computing*, Vol **28**(4): pages 1364-1396, 1999.
9. R. Impagliazzo, L.A. Levin and M. Luby, "Pseudo-random Generation from One-way Functions"; *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 12-24, 1989.
10. R. Impagliazzo, M. Naor, "Efficient Cryptographic Schemes Provably as Secure as Subset Sum"; *Journal of Cryptology*, **9**(4): pages 199-216, 1996.
11. M. Kharitonov, "Cryptographic Hardness of Distribution-specific Learning"; *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 372-381, 1993.
12. M. Kharitonov, A.V. Goldberg and M. Yung, "Lower Bounds for Pseudorandom Number Generators"; *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 242-247, 1989.
13. M. Krause and S. Lucks, "On the minimal Hardware Complexity of Pseudorandom Function Generators", *Proceedings of the 18th Symposium on Theoretical Aspects of Computer Science*, 2001.
14. N. Linial, Y. Mansour and N. Nisan, "Constant Depth Circuits, Fourier Transform, and Learnability", *Journal of the ACM*, Vol **40**(3): pages 607-620, 1993.
15. C. McDiarmid, "On the method of bounded differences", *London Mathematical Society Lecture Note Series* **141**, Cambridge University Press, 1989, 148–188.
16. M. Naor and O. Reingold, "Synthesizers and Their Application to the Parallel Construction of Pseudorandom Functions", *Journal of Computer and Systems Sciences*, **58**(2): pages 336-375, 1999.
17. C.H. Papadimitriou, *Computational Complexity*; Addison-Wesley, 1994.
18. A.C-C. Yao, "Theory and Applications of Trapdoor Functions"; *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 80-91, 1982.
19. X. Yu and M. Yung, "Space Lower-Bounds for Pseudorandom-Generators"; *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, pages 186-197, 1994.