General pseudo-random generators from weaker models of computation

George Karakostas
Dept. of Computing and Software
McMaster University, Hamilton, ON, Canada
gk@cas.mcmaster.ca
fax: +1 905-524-0340

April 16, 2003

Abstract

The construction of pseudo-random generators (PRGs) has been based on strong assumptions like the existence of one-way functions or exponential lower bounds for the circuit complexity of Boolean functions. Given our current lack of satisfactory progress towards proving these assumptions, we study the implications of constructing PRGs for weaker models of computation to the derandomization of general classes like *BPP*. More specifically, we show how PRGs that fool monotone circuits could lead to derandomization for general complexity classes, and how the Nisan-Wigderson construction could use hardness results for monotone circuits to produce pseudo-random strings.

Keywords: Pseudo-random generators, circuit complexity, monotone circuit complexity

1 Introduction

One of the central issues in computational complexity is the understanding of the the additional (if any) power randomization can offer to solving problems efficiently. The investigation of the exact relations between randomized computational classes like *BPP* and non-randomized ones (like *P, NP, EXP*) has been the subject of intense research for the last two decades. One of the major steps towards a better understanding of randomness in computation was the realization [15] that hardness results can lead to the construction of good pseudo-random generators. This observation led to the construction of pseudo-random generators based on the (assumed) intractability of the discrete logarithm function [3] or the existence of one-way functions [18], [7]. The seminal paper by Nisan and Wigderson [11] was the first to connect the existence of predicates with high circuit complexity to the existence of good pseudo-random generators. The hardness assumption of [11] was still very strong (although not as strong as previous assumptions, e.g., the existence of one-way functions): the predicate used by the generator had to be extremely hard *on average*. A series of results based on hardness amplification techniques like Yao's XOR lemma [18] or error-correcting codes [16] can be used in order to relax this requirement for an extremely hard function to constant hardness [9], to mildly hard [5], to worst-case hard functions [2].

Unfortunately, this spectacular progress towards weaker hardness assumptions for generators of a given power has not brought us close to resolving the still open issue of the power of randomness in computation. This is due to the bleak current state-of-the-art in lower bounds for general circuits. We are not even close to proving any of the hardness assumptions above, and if the past is an indication of the future, we shouldn't expect their proof any time soon. Motivated by these difficulties, we ask whether looking for explicit hard functions in the general circuit model in order to construct pseudo-random generators is really an overkill. Indeed, let's assume that we can show the following 'theorem': if there is a (general) circuit of a certain size that can distinguish a truly random string to a string produced by a generator that uses only a random seed with some probability, then there is a construction in a restricted model of computation (e.g., a monotone circuit) of a certain size that can distinguish the two strings with a somewhat smaller probability. The contrapositive of the above 'theorem' would imply that if we can construct a generator that fools *all* restricted constructions (e.g., monotone circuits) of a certain size, then this generator would fool *all* general circuits of a certain size with a somewhat smaller probability. We apply this general framework to the restricted model of monotone circuits (in fact circuits that compute slice functions). There has been great progress in proving strong unconditional lower bounds for monotone circuits, for example [14], [10], [13], [4]. Razborov's approximation technique has even been extended to general circuits with only a few NOT gates [1]. In Section 2 we formalize the intuition above as Theorem 1 and give its (almost trivial) proof. Then we study the Nisan-Wigderson construction as a generator for monotone circuits. The hardness requirements of this generator are much stronger than just worst-case hardness, and there are no strong enough hardness amplification techniques for monotone circuits yet (indeed, it may be the case that no such techniques exist). Hence the generator still isn't unconditionally pseudo-random. Two facts though allow us to be optimistic about this approach: The first is the tremendous success of proving lower bounds for monotone circuits. The second fact is that the Nisan-Wigderson construction was built with fooling general circuits in mind. Once we set a more modest goal (fooling monotone circuits only, for example), other constructions may do better in terms of hardness requirements. We discuss these issues together with other possible directions in what we consider to be the most important part of this work, Section 4.

2 Monotone circuit tests from general circuit tests

In what follows, $\mathcal{C}$ and $\mathcal{C}_M$ are the classes of (general) circuits and monotone circuits respectively.

Definition 1 *A function $G : \{0,1\}^d \to \{0,1\}^n$ is an* (s, ε) pseudo-random generator *if no circuit of size s can distinguish G from the uniform distribution U_n with advantage greater than ε. That is, for*

every circuit C of size at most s,

$$\Big|\Pr\big[C(U_n)=1\big]-\Pr\big[C(G(U_d))=1\big]\Big|\leq\varepsilon$$

The circuit C in the definition above can be thought as a test that the alleged PRG has to pass in order to be pseudo-random.

Lemma 1 *If there is circuit $C\in\mathcal{C}$ of size s such that*

$$\Big|\Pr_{y\in\{0,1\}^n}\big[C(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C(G(x))=1\big]\Big|>\varepsilon$$

then there is a monotone circuit $C_M\in\mathcal{C}_M$ of size $|C_M|=O(s+n\log^2 n)$ such that

$$\Big|\Pr_{y\in\{0,1\}^n}\big[C_M(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C_M(G(x))=1\big]\Big|>\frac{\varepsilon}{2(n+1)}$$

Proof: W.l.o.g. we will assume that

$$\Pr_{y\in\{0,1\}^n}\big[C(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C(G(x))=1\big]>\varepsilon$$

Let $y_i, G_i(x)$ be the i^{th} bit of $y, G(x)$. Then

$$\Pr_{y\in\{0,1\}^n}\big[C(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C(G(x))=1\big]=\sum_{k=0}^{n}\Bigg(\Pr_y\big[\sum_i y_i=k\big]\Pr_y\big[C(y)=1|\sum_i y_i=k\big]-$$

$$-\Pr_x\big[\sum_i G_i(x)=k\big]\Pr_x\big[C(G(x))=1|\sum_i G_i(x)=k\big]\Bigg)>\varepsilon$$

Therefore there is $0\leq k_0\leq n$ such that

$$\begin{aligned}&\Pr_y\big[\sum_i y_i=k_0\big]\Pr_y\big[C(y)=1|\sum_i y_i=k_0\big]-\\&-\Pr_x\big[\sum_i G_i(x)=k_0\big]\Pr_x\big[C(G(x))=1|\sum_i G_i(x)=k_0\big]>\frac{\varepsilon}{n+1}\end{aligned}\tag{1}$$

Let C_M be the following slice function of monotone circuit complexity $O(s+n\log^2 n)$ (see [17]):

$$C_M(y)=\begin{cases}0, & \text{if } \sum_i y_i<k_0\\ 1, & \text{if } \sum_i y_i>k_0\\ C(y), & \text{if } \sum_i y_i=k_0\end{cases}$$

Then we have

$$\begin{aligned}&\Pr_{y\in\{0,1\}^n}\big[C_M(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C_M(G(x))=1\big]=\\&=\Pr_y\big[\sum_i y_i<k_0\big]\Pr_y\big[C_M(y)=1|\sum_i y_i<k_0\big]-\Pr_x\big[\sum_i G_i(x)<k_0\big]\Pr_x\big[C_M(G(x))=1|\sum_i G_i(x)<k_0\big]+\\&+\Pr_y\big[\sum_i y_i>k_0\big]\Pr_y\big[C_M(y)=1|\sum_i y_i>k_0\big]-\Pr_x\big[\sum_i G_i(x)>k_0\big]\Pr_x\big[C_M(G(x))=1|\sum_i G_i(x)>k_0\big]+\\&+\Pr_y\big[\sum_i y_i=k_0\big]\Pr_y\big[C_M(y)=1|\sum_i y_i=k_0\big]-\Pr_x\big[\sum_i G_i(x)=k_0\big]\Pr_x\big[C_M(G(x))=1|\sum_i G_i(x)=k_0\big]\end{aligned}$$

From the definition of C_M we know that

$$\begin{aligned}\Pr_y\big[C_M(y)=1|\sum_i y_i<k_0\big] &= \Pr_x\big[C_M(G(x))=1|\sum_i G_i(x)<k_0\big]=0\\ \Pr_y\big[C_M(y)=1|\sum_i y_i>k_0\big] &= \Pr_x\big[C_M(G(x))=1|\sum_i G_i(x)>k_0\big]=1\end{aligned}$$

hence

$$\Pr_{y\in\{0,1\}^n}\big[C_M(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C_M(G(x))=1\big]=\Pr_y\big[\sum_i y_i>k_0\big]-\Pr_x\big[\sum_i G_i(x)>k_0\big]+$$

$$+\Pr_y\big[\sum_i y_i=k_0\big]\Pr_y\big[C_M(y)=1|\sum_i y_i=k_0\big]-\Pr_x\big[\sum_i G_i(x)=k_0\big]\Pr_x\big[C_M(G(x))=1|\sum_i G_i(x)=k_0\big]$$

Let

$$\begin{aligned}A &= \Pr_y\big[\sum_i y_i>k_0\big]-\Pr_x\big[\sum_i G_i(x)>k_0\big]\\ B &= \Pr_y\big[\sum_i y_i=k_0\big]\Pr_y\big[C_M(y)=1|\sum_i y_i=k_0\big]-\\ &- \Pr_x\big[\sum_i G_i(x)=k_0\big]\Pr_x\big[C_M(G(x))=1|\sum_i G_i(x)=k_0\big]\end{aligned}$$

If $|A|>\frac{\varepsilon}{2(n+1)}$ then the threshold function

$$T_{k_0+1}(y)=\begin{cases}0, & \text{if } \sum_i y_i\le k_0\\ 1, & \text{if } \sum_i y_i>k_0\end{cases}$$

has the desired properties, and the lemma holds. So assume that $|A|\le\frac{\varepsilon}{2(n+1)}$. Also, from (1) we know that $B>\frac{\varepsilon}{n+1}$, therefore

$$\Pr_{y\in\{0,1\}^n}\big[C_M(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C_M(G(x))=1\big]=A+B>\frac{\varepsilon}{2(n+1)}$$

The proof is the same in case

$$\Pr_{x\in\{0,1\}^d}\big[C(G(x))=1\big]-\Pr_{y\in\{0,1\}^n}\big[C(y)=1\big]>\varepsilon$$

□

From the lemma above we get the following theorem:

Theorem 1 *There is a constant $\gamma>0$ such that if*

$$\Big|\Pr_{y\in\{0,1\}^n}\big[C_M(y)=1\big]-\Pr_{x\in\{0,1\}}\big[C_M(G(x))=1\big]\Big|\le\frac{\varepsilon}{2(n+1)}$$

for all monotone circuits C_M with size $|C_M|\le\gamma s+\gamma n^2\log n$ for some $s>0$, then

$$\Big|\Pr_{y\in\{0,1\}^n}\big[C(y)=1\big]-\Pr_{x\in\{0,1\}^d}\big[C(G(x))=1\big]\Big|\le\varepsilon$$

for all circuits $C\in\mathcal{C}$ of size $|C|\le s$.

Notice that nowhere in the above did we demand that G be a special kind of ciruit (e.g. monotone). Also notice that we can strengthen Lemma 1 and Theorem 1 by replacing the monotone circuit C_M by a monotone *slice* function C_M. Therefore in order to construct PRG's for general circuits, it is enough to construct PRG's for monotone (slice) circuits.

3 The Nisan-Wigderson generator for monotone circuits

As an illustrative example, we demonstrate how one can use the Nisan-Wigderson PRG for monotone circuits. We cannot stretch enough the fact that this PRG was constructed in order to fool *general* Boolean circuits, and therefore it may very well *not* be the appropriate way to go. Nevertheless, at this point this is one of the most successful constructions (in terms of derandomization power) and a

very good example for pointing out some of the different issues one may encounter during the design of a PRG for *monotone* circuits. Again, we emphasize that in what follows we make assumptions some of which may possibly be proven false, but we make them nevertheless, since they help us in illustrating our ideas or they can be transformed to other assumptions which are not so easily proven/disproved.

The main difficulty for applying the Nisan-Wigderson(NW) construction as is, is our inability to use the standard conversion of a circuit that is a good distinguisher between a truly random sequence and the output of the NW generator, to a circuit that approximates a hard function. This standard conversion uses the XOR of the output of such a circuit with a random bit, to produce the approximator, but in a monotone setting we cannot simulate the XOR. Therefore we need to modify the hardness assumption used by the NW generator:

Assumption 1 *There is a monotone predicate* $P : \{0,1\}^l \to \{0,1\}$ *such that no* monotone *circuit of size* s *can compute* P *or* $\bar{P}$ *correctly on more than a fraction* $\frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$ *of the* 2^l *inputs (*n *is the number of (pseudo)-random bits we need).*[1]

This assumption differs from the usual hardness assumptions associated with the NW generator in two major points: it is an assumption about the *monotone* complexity of a (monotone) predicate P, and it is also an assumption on the *monotone* complexity of the *non-monotone* complement of P. Also notice that the hardness assumption is stronger as far as the fraction of correct answers is concerned: it is $\frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$ instead of $\frac{1}{2} + \frac{\varepsilon}{n}$. The need for these modifications will become apparent when we try to prove that the NW construction is indeed a PRG. First we describe the Nisan-Wigderson construction.

Initially, this construction produces a collection of sets with small intersections (called a *design*):

Lemma 2 *[11] For every* $l, n \in \mathbb{N}$ *there exists a family of sets* $S_1, \ldots, S_n \subset \{1, \ldots, d\}$ *such that*

1. $d = O(\frac{l^2}{\log n})$
2. $|S_i| = l, \ \forall i$
3. $|S_i \cap S_j| \leq \log n, \ \forall i \neq j$

Moreover, such a family can be computed by a deterministic TM in time poly$(n, 2^d)$*. In case* $l = C \log n$ *for some constant* $C > 0$*, there exists such a design with* $d = O(C^2 \log n)$ *that can be computed by a deterministic TM in time poly*(n)*.*

The generator in [11] uses the fact that if we have a uniformly distributed string x of length d, we can produce a family of substrings x_{S_i} with the above properties that will behave as independent when they are used as inputs to a hard predicate. The NW construction will output the string

$$NW^P_{l,n}(x) = P(x_{S_1})P(x_{S_2}) \ldots P(x_{S_n}).$$

Following the approach of [11] we show that under our Assumption 1 the NW construction is a PRG.

Theorem 2 *Under Assumption 1 the NW construction is a* $(\gamma s - O(n^2 \log n), \varepsilon)$ *PRG for some constant* $\gamma > 0$*.*

Proof: The proof of the theorem is virtually the same as in [11], but we repeat it here in order to illustrate new directions and open problems that arise from the use of a weaker model (monotone circuits).

We will assume that the NW construction is not a PRG with the required size-error parameters and will arrive at a contradiction. Since we assume that the NW construction is not an $(\gamma s -$

[1] As stated, this assumption refers to monotone functions that are unbiased over *all inputs*. In fact, we can relax it by just requiring them to be unbiased over a particular *distribution* on all inputs that can be easily sampled. For example, we may talk about Razborov's CLIQUE function and the inputs to it are picked from only the "good" and "bad" inputs of Razborov's proof with equal probability.

$O(n^2 \log n), \varepsilon)$ PRG (for some constant $\gamma > 0$ to be picked later), there is a (possibly non-monotone) circuit C of size $\gamma s - O(n^2 \log n)$ such that

$$\left|\Pr_{y\in\{0,1\}^n}\left[C(y)=1\right] - \Pr_{x\in\{0,1\}^d}\left[C(NW^P_{l,n}(x))=1\right]\right| > \varepsilon$$

From Lemma 1 and after an appropriate choice of γ above, we get that there is a *monotone* circuit C_M of size $s - O(n^2 \log n)$ such that

$$\left|\Pr_{y\in\{0,1\}^n}\left[C_M(y)=1\right] - \Pr_{x\in\{0,1\}^d}\left[C_M(NW^P_{l,n}(x))=1\right]\right| > \frac{\varepsilon}{2(n+1)}$$

We assume that

$$\Pr_{y\in\{0,1\}^n}\left[C_M(y)=1\right] - \Pr_{x\in\{0,1\}^d}\left[C_M(NW^P_{l,n}(x))=1\right] > \frac{\varepsilon}{2(n+1)} \tag{2}$$

If this is not the case, we *complement* C_M and we work with this complemented monotone circuit (which now is *non-monotone*). Later we will see that this will not affect our proof. Following closely previous proofs related to the NW construction, we define the following hybrid distribution on $\{0,1\}^n$:

Distribution D_i: The first i bits are the first i bits of $NW^P_{l,n}(x)$, where x is chosen uniformly over d-bit strings, and the rest $n-i$ bits $r_{i+1}, r_{i+2}, \ldots, r_n$ are chosen uniformly and independently at random.

Since D_0 is U_n and D_n is $NW^P_{l,n}(x)$, we have that there is an i such that

$$\Pr\left[C_M(D_i)=1\right] - \Pr\left[C_M(D_{i-1})=1\right] > \frac{\varepsilon}{2n(n+1)}$$

or if we expand it

$$\begin{aligned}\Pr_{x,r_{i+1},\ldots,r_n}\left[C_M(P(x_{S_1})\ldots P(x_{S_{i-1}})P(x_{S_i})r_{i+1}\ldots r_n)=1\right] - \\ -\Pr_{x,r_i,\ldots,r_n}\left[C_M(P(x_{S_1})\ldots P(x_{S_{i-1}})r_i r_{i+1}\ldots r_n)=1\right] > \frac{\varepsilon}{2n(n+1)}\end{aligned} \tag{3}$$

At this point, the proof of the pseudorandomness properties of the NW construction when the hardness of general circuits is used, utilizes a standard transformation of a distinguisher satisfying inequality (3) to a predictor [18]. By renaming r_i to b we get that

$$\Pr_{x,b,r_{i+1},\ldots,r_n}\left[C_M(P(x_{S_1})\ldots P(x_{S_{i-1}})b r_{i+1}\ldots r_n)\oplus b = P(x_{S_i})\right] > \frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$$

Using standard averaging arguments, we can fix $b, r_{i+1}, \ldots, r_n$ as well as the bits of x not in S_i while preserving this prediction probability. Unfortunately, if b is fixed to 1, this transformation doesn't work in our case if we had just assumed the hardness of P (and not $\bar{P}$ as well), because it results in a *non-monotone* circuit approximating P (destroying our contradiction argument).

We rename x_{S_i} to z and we notice that the values of P calculated by the NW construction depend only on $|S_i \cap S_j| \leq \log n, j \neq i$ bits of z. Therefore these values are *monotone* functions P_j of z. So, depending on the (fixed) value of b, we have that either

$$\Pr_z\left[C'_M(P_1(z)\ldots P_{i-1}(z)) = P(z)\right] > \frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$$

or

$$\Pr_z\left[\,\overline{C'_M}(P_1(z)\ldots P_{i-1}(z)) = P(z)\right] > \frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$$

for some monotone circuit C'_M. Notice that we will also arrive at this point in the case (2) doesn't hold.

Each P_j can be computed by a DNF of size $O(n \log n)$, and there are at most n such functions. If we plug in C'_M the monotone circuits for each P_j, then we get a monotone circuit C'_M of size at most $s - O(n^2 \log n) + O(n^2 \log n) = s$ (for an appropriate choice of the constant in the first big-O), and such that either

$$\Pr_z\left[C'_M(z) = P(z)\right] > \frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$$

or

$$\Pr_z\left[C'_M(z) = \overline{P}(z)\right] > \frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$$

In either case, this contradicts Assumption 1. □

4 Discussion and open problems

In this section we explore some new directions and open problems suggested by the use of hard monotone functions for the construction of PRGs for general circuits.

4.1 Hardness assumptions

- The first (obvious) open question is whether there is indeed a monotone predicate P with the hardness properties of Assumption 1. It seems plausible that if a monotone function is very hard to approximate, then its *non-monotone* complement is even harder to approximate with *monotone* circuits. Razborov's approximation technique has been extended to general circuits with a limited number of NOT gates [1] quite naturally. Hence it is conceivable that techniques which would prove hardness of approximation for monotone circuits can be extended to prove hardness of approximation for circuits with, say, only one NOT gate before their output. Assumption 1 can be reformulated as follows:

  Assumption 2 *There is a predicate $P : \{0,1\}^l \to \{0,1\}$ such that no monotone circuit or circuit with exactly one NOT gate right before its output of size s can compute P correctly on more than a fraction $\frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$ of the 2^l inputs (n is the number of (pseudo)-random bits we need).*

- The monotonicity of P was necessary so that the circuit we get after plugging in the DNF for each P_j is still monotone. But we can modify Assumption 1 as follows:

  Assumption 3 *There is a (general) predicate $P : \{0,1\}^l \to \{0,1\}$ such that no monotone circuit of size s can compute P or $\overline{P}$ correctly on more than a fraction $\frac{1}{2} + \frac{\varepsilon}{2n(n+1)}$ of the 2^l inputs, using as advice the output of oracles for P that have $l - \log n$ input bits fixed (n is the number of (pseudo)-random bits we need).*

  Assumption 3 is a version of strong non-self-reducibility that trades the generalization of the circuit family for which we need strong lower bounds (monotone with advice instead of just monotone) for the generalization of the hard predicate used in the NW construction.

How hard is it to design PRGs for monotone circuits? Impagliazzo, Shaltiel and Wigderson [8] showed that derandomizing BPP using a pseudo-random generator implies that EXP $\not\subset$ P/poly. Impagliazzo and Kabanets [6] also show that derandomization of RP or BPP would imply superpolynomial lower bounds for Boolean or arithmetic circuits. Since the assumptions used above imply the construction of pseudo-random generators, their proof would immediately imply strong circuit lower bounds. Therefore proving any of our assumptions (provided any of them is true) is *as difficult as* proving such lower bounds. Notice though that our assumptions are about a *weaker* model than general boolean circuits, and in this model the development of (worst case) lower bounds has been very successful so far.

4.2 Hardness amplification for monotone predicates

Unfortunately the XOR function is not monotone, therefore Yao's XOR lemma cannot be applied directly in order to amplify the hardness of a function in a way useful for Theorem 2. Nevertheless, there are already some (weak) hardness amplification results for monotone functions. The recent work by O'Donnell [12] is in fact an amplification that applies to monotone circuits and their complements, in just the way we need it for Assumption 1. Instead of the XOR of several copies of a function, [12] uses two monotone functions: first it uses the $REC-MAJ-3_l$ function (l is the depth of a ternary tree of majority-of-3's) of several copies of P to go from an $(1-1/poly(n))$-hard predicate P for *general* polynomial-size circuits to a new $(1/2+o(1))$-hard predicate for polynomial circuits, and then it uses the Tribes function of Ben-Or and Linial, to transform the new predicate to a new $(1/2+1/n^{-1/2+\varepsilon})$-hard predicate for polynomial circuits. The definitions of the $REC-MAJ-3_l$ and Tribes functions can be found in [12]. An important property of these functions is that $REC-MAJ-3_l(\bar{P},\ldots,\bar{P}) = \overline{REC-MAJ-3_l}(P,\ldots,P)$ and $\text{Tribes}(\bar{P},\ldots,\bar{P}) = \overline{\text{Tribes}}(P,\ldots,P)$, therefore the construction can use a mild version of Assumption 1 to produce a harder version of this same assumption. The starting point of this amplification is Impagliazzo's hard-core set theorem [5], which holds with respect to any model of computation closed under majority (therefore it also holds for our model of computation with respect to which we assume hardness of some predicate, namely monotone circuits). But O'Donnell's construction analysis seems to be an overkill for our case, since we only need hardness against *monotone* circuits of some size. Nevertheless, this amplification method probably doesn't achieve the power of Yao's XOR lemma. The existence of powerful amplification techniques for monotone predicates remains an interesting open problem.

Acknowledgements

I would like to thank Sanjeev Arora for introducing me to the field of pseudorandomness, Dick Lipton for many enlightening discussions on the complexity of monotone functions, and Tasos Viglas, Iannis Tourlakis, and Nicola Galesi for reading an earlier draft of this work and for many helpful discussions. Many thanks to Valentine Kabanets for pointing [8] and [6] to me.

References

[1] K. Amano and A. Maruoka. Potential of the approximation method. 37^{th} *FOCS*, pp. 431-440, 1996.

[2] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. *BPP* has subexponential time simulations unless *EXPTIME* has publishable proofs. Computational Complexity, 3(4):307-318, 1993.

[3] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. on computing*, 13(4), pp. 850-864, 1984.

[4] D. Harnik and R. Raz. Higher Lower Bounds for Monotone Size. 32^{nd} *STOC*, pp. 191-201, 2000.

[5] R. Impagliazzo. Hard-core distributions for somewhat hard problems. 36^{th} *FOCS*, 1995.

[6] R. Impagliazzo and V. Kabanets. Derandomizing Polynomial Identity Tests means proving circuit lower bounds. 35^{th} *STOC*, (to appear), 2003.

[7] R. Impagliazzo, L. Levin, and M. Luby. Pseudorandom generators from any one-way function. 21^{st} *STOC*, 1989.

[8] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Near-optimal conversion of hardness into pseudo-randomness. 40^{th} *FOCS*, pp. 181-190, 1999.

[9] R. Impagliazzo and A. Wigderson. *P=BPP* if E requires exponential circuits. 29^{th} *STOC*, pp. 220-229, 1998.

[10] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. on Discrete Mathematics*, 3(2), pp. 255-265, 1990.

[11] N. Nisan and A. Wigderson. Hardness vs. Randomness. *JCSS*, vol. 49, No. 2, pp. 149-167, Oct. 1994.

[12] R. O'Donnell. Hardness Amplification Within NP. 34^{th} *STOC*, pp. 751-760, 2002.

[13] R. Raz and P. McKenzie. Separation of the Monotone NC Hierarchy. *Combinatorica* 19(3), pp. 403-435, 1999.

[14] A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Dokl. Akad. Nauk. SSSR*, 281(4), pp. 598-607, 1985. (In Russian)

[15] A. Shamir. On the generation of cryptographically strong pseudo-random sequences. 8^{th} *ICALP*, 1981.

[16] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. 31^{st} *STOC*, pp. 537-546, 1999.

[17] I. Wegener. *The complexity of Boolean functions.* John Wiley, 1987.

[18] A.C. Yao. Theory and applications of trapdoor functions. 23^{rd} *FOCS*, pp. 80-91, 1982.