Derandomizing Arthur-Merlin games using hitting sets

Peter Bro Miltersen
Department of Computer Science
University of Aarhus, Denmark
bromille@daimi.au.dk.

N. V. Vinodchandran *
Department of Computer Science and Engineering
University of Nebraska-Lincoln, Lincoln, USA
vinod@cse.unl.edu

September 28, 2004

Abstract

We prove that **AM** (and hence Graph Nonisomorphism) is in **NP** if for some $\epsilon > 0$, some language in **NE** $\cap$ **coNE** requires nondeterministic circuits of size $2^{\epsilon n}$. This improves results of Arvind and Köbler and of Klivans and van Melkebeek who proved the same conclusion, but under stronger hardness assumptions.

The previous results on derandomizing **AM** were based on pseudorandom generators. In contrast, our approach is based on a strengthening of Andreev, Clementi and Rolim's hitting set approach to derandomization. As a spin-off, we show that this approach is strong enough to give an easy proof of the following implication: for some $\epsilon > 0$, if there is a language in **E** which requires nondeterministic circuits of size $2^{\epsilon n}$, then **P**=**BPP**. This differs from Impagliazzo and Wigderson's theorem "only" by replacing deterministic circuits with nondeterministic ones.

1 Introduction

Using hardness for simulating randomness has been a fundamental idea in complexity theory. The main objective is to find nontrivial deterministic simulations of an entire class of randomized algorithms (rather than just a specific one) under certain complexity theoretic hardness assumptions. Typically, the assumptions are in the form of the existence

*Research was done while this author was a postdoctoral fellow at BRICS, Department of Computer Science University of Aarhus, Denmark of functions in a uniform complexity class (for example **EXP**) that cannot be computed or approximated by a certain non-uniform class (for example polynomial size circuits). An early seminal result is the following result of Nisan and Wigderson that was proved by constructing a *pseudorandom generator*.

Theorem 1 (Nisan-Wigderson) *Let $\epsilon > 0$ be any constant. If there exists a language L in $\mathbf{E}$ so that for all but finitely many n, any circuit of size $2^{\epsilon n}$ agrees with the characteristic function of $L \cap \{0,1\}^n$ on at most a $\frac{1}{2} + 2^{-\epsilon n}$ fraction of $\{0,1\}^n$, then $\mathbf{P} = \mathbf{BPP}$.*

The hardness assumption in Theorem 1 is "average-case" rather than worst case. Substantial research has been done in order to remedy this and arguably the most remarkable result is a theorem due to Impagliazzo and Wigderson [IW97]. They showed the following improvement of Theorem 1.

Theorem 2 (Impagliazzo-Wigderson) *Let $\epsilon > 0$ be any constant. If there exists a language L in $\mathbf{E}$ so that for all but finitely many n, $L \cap \{0,1\}^n$ has circuit complexity at least $2^{\epsilon n}$, then $\mathbf{P}$=$\mathbf{BPP}$.*

The proof of this theorem is technical and is built on the results of many earlier papers, including [BM84, Yao82, NW94, GL89, BFNW93, Imp95]. This result has been subsequently simplified and extended to get derandomization results for a range of parameters [STV01, ISW99, ISW00, SU01, Uma03].

Although much research has gone into derandomizing **BPP** and **RP**, derandomization of classes like **AM** has received attention only recently. The class **AM** was defined, by Babai, and Babai and Moran in [Bab85, BM88], as a natural randomized (and interactive) version of the class **NP**. A number of natural computational problems have been shown to be in **AM** but are not known to be in **NP** [Bab85, BM88, GMW91, GS89, Bab92]. Most have a group theoretic flavor. The most celebrated one among them is the Graph Nonisomorphism problem. A complete derandomization of **AM** (that is, a proof of the statement **AM**=**NP**) would immediately give polynomial size membership proofs for positive instances of Graph Nonisomorphism. In contrast, the lengths of the shortest proofs known, without any assumptions, are exponential in the sizes of the graphs [BL83, BKL83].

In [AK01], Arvind and Köbler showed that the construction of [NW94] can be extended to the nondeterministic setting to get pseudorandom generators which can be used to completely derandomize **AM**. As in the case of [NW94], they needed an average-case hardness assumption in order to construct the generator. To be precise, Arvind and Köbler showed the following theorem.

Theorem 3 (Arvind-Köbler) *Let $\epsilon > 0$ be any constant. If there exists a language L in $\mathbf{NE} \cap \mathbf{coNE}$[1] so that for all but finitely many n any nondeterministic circuit of size $2^{\epsilon n}$ agrees with the characteristic function of $L \cap \{0,1\}^n$ on at most a $\frac{1}{2} + 2^{-\epsilon n}$ fraction of $\{0,1\}^n$, then $\mathbf{AM} = \mathbf{NP}$.*

[1] Arvind and Köbler only state the theorem under the assumption $L \in \mathbf{E}$, but their proof easily generalizes.

Klivans and van Melkebeek [KvM02] constructed generators for derandomizing **AM** under a worst-case hardness assumption. The main observation they make is that the proof of Impagliazzo and Wigderson *relativizes*. This leads to the following theorem.

Theorem 4 (Klivans-van Melkebeek) *Let $\epsilon > 0$ be any constant. If there exists a language L in* **NE** $\cap$ **coNE** *so that for all but finitely many n, $L \cap \{0,1\}^n$ has oracle circuit complexity at least $2^{\epsilon n}$ with oracle gates for* SAT*, then* **AM**=**NP**.

Here, *oracle circuits* are Boolean circuits which contain special gates called *oracle* gates. These oracle gates are of unbounded fanin (but a gate of fan-in r contributes size r to the circuit) and can be used for oracle access to a language, in this case SAT. The output of the gate on a string x is 1 if $x \in$ SAT. Otherwise the output is 0.

Arvind and Köbler [AK01] and van Melkebeek [vM98] asked whether **AM**=**NP** follows from the existence of a language in **NE**$\cap$**coNE** which does not have subexponential nondeterministic circuit complexity. In this paper, we answer this question affirmatively, proving the following theorem which improves Theorem 3 as well as Theorem 4.

Theorem 5 *Let $\epsilon > 0$ be any constant. If there exists a language L in* **NE** $\cap$ **coNE** *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$, then* **AM**=**NP**.

Here an SV (single valued) nondeterministic circuit is a restriction of the notion of a nondeterministic circuit: in an SV-nondeterministic circuit, there are two output bits, the real output bit, and a flag, indicating whether the computation has been correctly performed. On both positive and negative instances, if the flag is on, the output bit should be correct. Additionally, for all instances, there should be some setting of the nondeterministic choice bits that make the flag turn on.

To see the difference between our result and the result of Klivans and van Melkebeek, we can informally say that SV-nondeterministic circuits of the stated size form a non-uniform and exponential analogue of **NP** $\cap$ **coNP**, while oracle circuits with SAT as an oracle form a non-uniform and exponential analogue of $\mathbf{P}^{\mathbf{NP}}$ (see Section 7 for a more detailed discussion on this issue).

Our approach to proving Theorem 5 is completely different from the techniques of Arvind and Köbler and of Klivans and van Melkebeek. Instead of using pseudorandom generators, we use a strengthened version of the *hitting set generator* approach to derandomization, due to Andreev, Clementi and Rolim [ACR97]. They gave, independently and almost simultaneously to Impagliazzo and Wigderson's work, two different conditions, each implying **P**=**BPP**. The conditions were much stronger than the hardness assumption in the Impagliazzo-Wigderson theorem; one of them essentially stating that there should be an algorithm operating in time polynomial in the size of its output, which on input n, m outputs the truth table of a Boolean function f from $\{0,1\}^n$ to $\{0,1\}^m$ with circuit complexity within a certain additive low order term of the maximum possible.

Their proof had two parts. First it is shown that the stated condition implies the existence of a certain hitting set generator (for definition of hitting set, see Section 2).

Then it is shown that the existence of such a generator implies **P**=**BPP** (it is easy to show that it implies **P**=**RP**). The latter part of the proof, i.e., the fact that the existence of the hitting set generator is enough to show **P**=**BPP** was proved already by Andreev, Clementi and Rolim in 1995 [ACR96b, ACR98]. Since then, the proof of this implication was simplified enormously [ACRT97, BF99, GVW00].

It was not (and is still not) clear if the hitting set approach to derandomization can be pushed to yield the Impagliazzo-Wigderson theorem. However, in this paper, we show, by strengthening the first part of their proof, that it can be pushed to yield the following statement.

Theorem 6 *Let $\epsilon > 0$ be any constant. If there exists a language L in* **E** *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$, then* **P**= **BPP**.

Note that this differs from the Impagliazzo-Wigderson theorem "only" in the assumption being about SV-nondeterministic circuits, rather than about deterministic ones.

Our main technical result is the following theorem, describing a procedure for turning the truth table of a Boolean function with big circuit complexity into a hitting set for circuits with very high acceptance probability (for precise definitions of the terms in the theorem, we refer the reader to Section 2).

Theorem 7 *For any $\epsilon > 0$, and $q \geq 1$, there is a polynomial time procedure P with the following properties. Let $f : \{0,1\}^m \rightarrow \{0,1\}$ be a function that cannot be computed by SV-nondeterministic circuits of size less than $2^{\epsilon m}$ for almost all m. Then there are constants $\delta = \delta(\epsilon) < \epsilon$ and $k > q$ such that, given the truth-table of $f : \{0,1\}^{kl} \rightarrow \{0,1\}$ as input, P outputs a hitting set $H_f \subseteq \{0,1\}^n$ for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+n^{\delta}}$, where $n = (2l)2^{2l}$.*

The main ingredient we add to the techniques of [ACR97] to prove Theorem 7 is to first replace the truth-table of f by a *multidimensional* encoding of f using an appropriate error-correcting code. An intuitive reason why this turns out to be useful is as follows. The technique of [ACR97] is based on *compression* in the form of hashing. As was previously noted in [Mil98], hashing becomes a much easier and cleaner operation when applied to data encoded in an error-correcting code. This essentially enables us to compress a *multi-dimensional* object along *all* dimensions, rather than just compressing a *two-dimensional* object along *one* dimension, as done in [ACR97]. We use *low degree extension* [BFLS91] of f for encoding purposes.

While the above intuition was useful for coming up with the proof of Theorem 7, the self-contained proof we present in Section 3 is quite short and the above intuition should not be necessary for understanding it.

Having proven Theorem 7, we combine it with a variation of a lemma from [ACR96a] (Lemma 13 of the present paper), and prove the following.

Corollary 8 *For any constant $\epsilon > 0$, there is a constant $\tau > 0$ so that the following holds. There is a deterministic polynomial time procedure which, given as input the truth table of a Boolean function $f : \{0,1\}^m \to \{0,1\}$ (i.e., 2^m bits) with SV-nondeterministic circuit complexity at least $2^{\epsilon m}$, outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n, where $n = \lceil 2^{\tau m} \rceil$.*

This corollary then immediately implies Theorem 5. To prove Theorem 6, we use the result of [ACR98], stating that the hitting set generator of Corollary 8 derandomizes **BPP**.

The existence of explicit dispersers is the main technical tool that we have to employ in order to prove Corollary 8. But, we would like to point out that any *relativizable* proof of Corollary 8 (such as ours) *has* to use the existence of explicit dispersers. More precisely, we note that any relativizable worst case hardness-based hitting set generator defines a disperser. The truth of this statement can be seen by arguing along the lines of [Tre01] where an analogous statement for extractors is implicitly established. We formally state this hitting set/disperser correspondence as Theorem 20 in Section 4.

In Section 5, we note that the techniques used in this paper to prove our main derandomization result can be used to show other hardness-randomness tradeoffs for **AM**. More precisely, we show the following theorem.

Theorem 9 *Let $\epsilon > 0$ be any constant. If there exists a language L in* **NEXP**$\cap$**coNEXP** *exists so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{n^{\frac{1}{2}+\epsilon}}$, then* **AM** $\subseteq$ **NQuasiP**.

The results in this paper were first published in 1999 [MV99]. Since then there has been significant progress in derandomizing **AM**. In Section 7, we briefly discuss these results.

2 Terminology and Preliminary Results

Lower case Greek letters denote rational constants between 0 and 1. The symbol log denotes $\log_2$.

Complexity classes

We assume standard textbook [BDG95, Pap94] complexity theoretic notation and definitions such as the definitions of standard complexity classes like **P**, **NP**, **E**, **NE**, **NEXP** and **BPP**. Here we only give the definition of the class **AM**.

A language L is defined[2] to be in **AM** if there is a language $L' \in$ **P** and a polynomial p, so that for all $x \in \{0,1\}^n$,

$$x \in L \Rightarrow \Pr_{y \in \{0,1\}^{p(n)}} (\exists z \in \{0,1\}^{p(n)} \ (x, y, z) \in L') = 1$$

[2] The original definition in [Bab85] of **AM** is a two-sided error version. But it is shown in [FGM+89] that this definition is equivalent to the one-sided error version, which we give here.

$$x \notin L \Rightarrow \Pr_{y \in \{0,1\}^{p(n)}} (\exists z \in \{0,1\}^{p(n)}\ (x, y, z) \in L') \leq \frac{1}{2}$$

An SVNP-procedure (SV meaning *Single Valued* [Sel96]) computing a function f is a polynomial time nondeterministic procedure so that every computation path on input x either produces $f(x)$ or rejects. Furthermore, at least one computation path must produce $f(x)$.

Circuits

A *nondeterministic* Boolean circuit C contains, in addition to AND, OR and NOT gates, *choice*-gates of fan-in 0. The circuit evaluates to 1 on an input x, and we say that $C(x) = 1$, if there is some assignment of truth values to the choice-gates that makes the circuit evaluate to 1. Otherwise $C(x) = 0$. A *co-nondeterministic* circuit C is defined similarly: the circuit evaluates to 0 on an input x, and we say that $C(x) = 0$, if there is some assignment of truth values to the choice-gates that makes the circuit evaluate to 0. Otherwise $C(x) = 1$.

Similarly, an SV-*nondeterministic* circuit C computing a function f has, in addition to its usual output, an extra output bit called the *flag*. For any input x and any setting of the choice-gates, if the flag is on, the circuit should output the correct value of $f(x)$. Furthermore, for any x, there should be some setting of the choice-gates that turn the flag on. It is easy to see that a Boolean function f has an SV-nondeterministic circuit of size $O(s(n))$ if and only if f has a nondeterministic circuit of size $O(s(n))$ and a co-nondeterministic circuit of size $O(s(n))$.

Oracle circuits [Wil85] are Boolean circuits with special gates called *oracle* gates. These oracle gates can be of arbitrary fanin, though a gate of fan-in r contributes size r to the circuit and can be used for oracle access to a fixed language, say L. The output of the gate on a string x is 1 if $x \in L$, otherwise the output is 0. Nondeterministic and SV-nondeterministic oracle circuits are defined by combining the above definitions in the obvious way.

Dispersers

For the purposes of this paper (there are more parameters in the general definition), a *disperser* with threshold t is a bipartite graph $G = (U, V, E)$ such that, for any subset $S \subseteq U$ with $|S| \geq t$, more than half the vertices of V are adjacent to S.

Also for the purposes of this paper, for constants $\delta, \gamma > 0$ and $k \geq 1$, an *explicit* (n^δ, n^γ)-disperser is a family of dispersers $G_n = (U_n, V_n, E_n), n = 1, 2, \ldots$ with $|U_n| = \{0,1\}^n, |V_n| = \{0,1\}^{\lceil n^\gamma \rceil}$, and threshold $t_n = 2^{n^\delta}$ so that there is a deterministic polynomial time algorithm which on input $x \in U_n$ enumerates the vertices in V_n adjacent to x (in particular, the outdegree of every $x \in U_n$ must be polynomial).

The first construction of explicit dispersers with these parameters was given by Saks, Srinivasan and Zhou in [SSZ98]. Their construction was subsequently simplified and improved in several papers including [TS02, Tre01, TSUZ01]. For an excellent exposition on recent developments in the construction of explicit extractors and dispersers we refer the reader to the survey paper by Shaltiel [Sha02]. For the theorem below, the original result by Saks, Srinivasan and Zhou suffices.

Theorem 10 (Saks-Srinivasan-Zhou) *For any $\delta > 0$, there is a $\gamma > 0$ so that an explicit (n^δ, n^γ)-disperser exists.*

Hitting sets

A *hitting set* in $\{0,1\}^n$ with threshold $\delta(n)$ for co-nondeterministic circuits of size $s(n)$ is a subset H of $\{0,1\}^n$ so that for any co-nondeterministic circuit C of size $s(n)$, taking n inputs and producing one output, the following holds: if $\Pr_{x\in\{0,1\}^n}[C(x) = 1] \geq \delta(n)$, then $\exists x \in H, C(x) = 1$. (The more usual definition of hitting sets for deterministic circuits is analogous).

With this definition, the following proposition is easy to prove.

Proposition 11 *If there is an SVNP-procedure which on input 1^n outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n, then* $\mathbf{AM} = \mathbf{NP}$.

Now we state a lemma from [ACR96a][3]. Actually, the lemma is already implicit in [Sip88]. It shows that in fact it is sufficient to construct hitting sets with much bigger threshold than $\frac{1}{2}$. This lemma is a consequence of the existence of explicit dispersers. Indeed, in [Sip88], it was Sipser's motivation for defining the notion of a disperser.

Lemma 12 (Sipser, Andreev-Clementi-Rolim) *For any constant $\delta > 0$, there are constants $q \geq 1$ and $\gamma > 0$ so that the following holds. There is a polynomial time procedure which, on input H where H is a hitting set in $\{0,1\}^n$ with threshold $1 - 2^{-n+n^\delta}$ for circuits of size n^q, outputs a hitting set in $\{0,1\}^{n'}$ with threshold $\frac{1}{2}$ for circuits of size n', where $n' = \lceil n^\gamma \rceil$.*

What we actually need is the analogous Lemma for co-nondeterministic circuits. This lemma is proved exactly as Lemma 12 using explicit dispersers. To make the paper self-contained, we give the proof. In the proof, for a circuit C, let $Z(C)$ denote the set of instances for which C evaluates to 0.

Lemma 13 *For any constant $\delta > 0$, there are constants $q \geq 1$ and $\gamma > 0$ so that the following holds. There is a polynomial time procedure which, on input H where H is a hitting set in $\{0,1\}^n$ with threshold $1 - 2^{-n+n^\delta}$ for co-nondeterministic circuits of size n^q, outputs a hitting set in $\{0,1\}^{n'}$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n', where $n' = \lceil n^\gamma \rceil$.*

[3]The reader should note that the lemma can only be found in the *revised* version of the cited ECCC technical report.

Proof Let $\delta > 0$ be fixed. According to Theorem 10, there is a γ so that an explicit (n^δ, n^γ)-disperser exists. Let $G_n = (U_n, V_n, E_n)$ be this disperser, i.e., with $n' = \lceil n^\gamma \rceil$, $U_n = \{0,1\}^n$, $V_n = \{0,1\}^{n'}$, and for all subsets S of U_n of size at least 2^{n^δ}, more than half the vertices of V are adjacent to S.

Let $H \subseteq \{0,1\}^n$ be a hitting set with threshold $1 - 2^{-n+n^\delta}$ for co-nondeterministic circuits of size n^q, where the constant q will be determined below.

Note that H is a subset of U_n. Let H' be the set of vertices in V_n adjacent to H. As the disperser is explicit, H' can be generated in polynomial time from H. We claim that it is a hitting set with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n'. Once we show this claim, we are done.

Indeed, take any co-nondeterministic circuit C' of size n' with n' inputs so that $|Z(C')| \leq \frac{2^{n'}}{2}$. We have to show that H' is not a subset of $Z(C')$. For this, construct a co-nondeterministic circuit C with n inputs as follows: $C(x) = 1$ iff $\exists y, (x, y) \in E_n \wedge C'(y)$. As the disperser is explicit, the size of this circuit can be made polynomial. We fix the constant q, so that n^q is an upper bound on its size. We claim $|Z(C)| < 2^{n^\delta}$: Otherwise, as G_n is a disperser, the neighbours in V_n of $Z(C)$ are more than half of V_n and thus the neighbours must intersect $V_n - Z(C')$, i.e., for some y adjacent to $x \in Z(C)$, $C'(y)$ is 1. But this implies $C(x) = 1$, contradicting $x \in Z(C)$. As $|Z(C)| < 2^{n^\delta}$, the acceptance probability of C is at least $1 - 2^{-n+n^\delta}$. Thus, as H is a hitting set, for some value $x \in H$, $C(x) = 1$. This means that for some $y \in V_n$, adjacent to some $x \in H$, $C'(y) = 1$. But such a y is by definition in H', i.e., H' hits C' as was to be shown. ∎

3 Simulating AM in NP

In this section we prove Theorem 5. We first prove Theorem 7 (stated in the Introduction) which shows how to construct a hitting set from the truth-table of a hard function. We restate this theorem below.

Theorem 7 *For any $\epsilon > 0$, and $q \geq 1$, there is a polynomial time procedure P with the following properties. Let $f : \{0,1\}^m \to \{0,1\}$ be a function that cannot be computed by SV-nondeterministic circuits of size less than $2^{\epsilon m}$ for almost all m. Then there are constants $\delta = \delta(\epsilon) < \epsilon$ and $k > q$ such that, given the truth-table of $f : \{0,1\}^{kl} \to \{0,1\}$ as input, P outputs a hitting set $H_f \subseteq \{0,1\}^n$ for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+n^\delta}$, where $n = (2l)2^{2l}$.*

Proof We first show how to efficiently generate the set $H_f \subseteq \{0,1\}^n$ from the truth table for f and then argue that it has the right property. In the following, the values of k and δ, which depend on ϵ and q, will be fixed later. We assume, without loss of generality, that l is sufficiently large.

View f as a map $f : (\{0,1\}^l)^k \to \{0,1\}$. Now let $\mathbf{F}$ be the finite field with 2^{2l} elements. Identify $\mathbf{F}$ with $\{0,1\}^{2l}$ in any way that makes arithmetic efficient and embed $\{0,1\}^l$ in $\mathbf{F}$ by padding with zeros. Let the *low degree extension* [BFLS91] $\tilde{f} : \mathbf{F}^k \to \mathbf{F}$ of f be the unique polynomial with individual degree in each variable at most $2^l - 1$, agreeing with f on $(\{0,1\}^l)^k$.

Informally speaking, we define H_f as the set of tabulations of the restrictions of $\tilde{f}$ to every axis-parallel line in $\mathbf{F}^k$. More precisely, for $i \in \{1, \ldots, k\}$ and $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k \in \mathbf{F}$, let $v_i(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ be the vector $(w_j)_{j \in \mathbf{F}}$ in $\mathbf{F}^{2^{2l}}$, with $w_j = \tilde{f}(a_1, \ldots, a_{i-1}, j, a_{i+1}, \ldots, a_k)$. As we have identified $\mathbf{F}$ with $\{0,1\}^{2l}$, we can also view $v_i(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ as a bit string in $\{0,1\}^{(2l)2^{2l}} = \{0,1\}^n$.

With this in mind, now define $H_i \subseteq \{0,1\}^n$ as follows.

$$H_i = \{v_i(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k) | a_1, \ldots, a_k \in \mathbf{F}\}$$

H_f is the union of all the H_is. That is,

$$H_f = \bigcup_{i=1}^{k} H_i.$$

First note that generating H_f from the truth table of f is a polynomial time procedure.

The structure of the proof that H_f is a hitting set with the desired properties is the following. We will suppose to the contrary that H_f is not such a hitting set, i.e., that it does not hit some circuit C. We will then show that f has a smaller SV-nondeterministic circuit than it is assumed to have. This will be done by making a *compressed* representation of $\tilde{f}$ which will have enough information to efficiently evaluate $\tilde{f}$ (and hence f) at any given point. The compressed representation is a table of the restriction of $\tilde{f}$ to S^k, for a small subset S of $\mathbf{F}$. The set S is carefully chosen, depending on the circuit C. Using the circuit C, we will be able to reconstruct $\tilde{f}$ on any desired point in $\mathbf{F}^k$ from its values in S^k.

Now we give a formal proof. Let $S \subseteq \mathbf{F}$ be a set of indices. Let π_S denote the projection function from any set of vectors in $\mathbf{F}^{|\mathbf{F}|}$ to $\mathbf{F}^S$.

We need to find a subset $S \subseteq \mathbf{F}$, so that the restriction of $\tilde{f}$ to S^k can be used to reconstruct $\tilde{f}$ on any desired point. The following lemma is used for finding such a set. The proof of the lemma uses the *error correcting* properties of low-degree polynomials.

Lemma 14 *Let $\mathbf{F}$ be a finite field of size greater than 2. For any polynomial p, let L_p denote the vector $(p(i))_{i \in \mathbf{F}}$ in $\mathbf{F}^{|\mathbf{F}|}$. Let $\mathcal{L} = \{L_p \mid p$ is a polynomial of degree $\leq |\mathbf{F}|^{\frac{1}{2}}\}$. Then for any set $Z \subset \mathbf{F}^{|\mathbf{F}|}$, there is a set of indices $S \subset \{1, \ldots, |\mathbf{F}|\}$, $|S| \leq \lceil log|Z| \rceil$ such that the projection $\pi_S : \mathcal{L} \cap Z \to \mathbf{F}^S$ is 1-1.*

Proof For any $x, y \in \mathcal{L}$, $x \neq y$ are same on less than $|\mathbf{F}|^{\frac{1}{2}}$ indices. Since $|\mathbf{F}| > 2$, this is less than $\frac{1}{4}$ fraction of all the indices. Let $r = \lceil log|Z| \rceil$. We can construct $S = \{j_1, \ldots, j_r\}$ in a greedy manner. Having chosen indices $S_i = \{j_1, \ldots j_i\}$, we construct $S_{i+1} = \{j_{i+1}\} \cup S_i$ as follows.

Let Y_i be the set of distinct pairs (x, y) such that x and y coincide on S_i. That is, $Y_i = \{(x,y) | x, y \in \mathcal{L} \cap Z,\ x \neq y,$ and $\pi_{S_i}(x) = \pi_{S_i}(y)\}$. For any fixed $(x, y) \in Y_i$, probability that x and y are different on a random index j in $\{1, \ldots, |\mathbf{F}\} - S_i$ is at least $\frac{3}{4}$. Hence, the expected number of pairs $(x, y) \in Y_i$ so that $x_j \neq y_j$ is at least $\frac{3|Y_i|}{4}$. So, by an averaging argument, we can find an index j_{i+1} such that $|Y_{i+1}| \le \frac{|Y_i|}{4}$. Since $\binom{|Z|}{2}/4^r < 1$, the lemma follows. ■

Now assume H_f is not the desired hitting set. Let C be a co-nondeterministic circuit establishing this, i.e., C maps $\{0,1\}^n$ to $\{0,1\}$, it has size n^q, and if we by Z denote $\{x \in \{0,1\}^n | C_i(x) = 0\}$, i.e., those x for which there is some setting of the nondeterministic choice gates making C evaluate to 0 on x, then $|Z| \le 2^{n^\delta}$ and $H_f \subseteq Z$.

Let $\mathcal{L} \subseteq \mathbf{F}^{2^{2l}}$ be the vectors of the form $(p(i))_{i \in \mathbf{F}}$ for some univariate polynomial of degree less than 2^l. We can also view $\mathcal{L}$ as a subset of $\{0,1\}^n$. By construction, $H_f \subseteq \mathcal{L}$. Also by assumption $H_f \subseteq Z$. Hence by Lemma 14, there is a set S of indices of size $\le n^\delta$ so that $\pi_S : \mathcal{L} \cap Z \to \mathbf{F}^S$ is 1-1. Fix this set S.

We will now construct a small SV-nondeterministic circuit for f. In fact, we will exhibit an efficient SV-nondeterministic procedure computing $\tilde{f}$ (and hence f) with the following non-uniform advice: the circuit C, the set S, and a table of the restriction of $\tilde{f}$ to S^k.

An overview of the procedure is the following. We need to compute $\tilde{f}$ on an arbitrary input $(a_1, a_2, \ldots, a_k)$. This is done in k stages. Let T_i denote the table of values $\tilde{f}(a_1, \ldots, a_i, S^{k-i})$. Then $T_k = \tilde{f}(a_1, a_2, \ldots, a_k)$ is the value that we want to compute and $T_0 = \tilde{f}(S^k)$ is the table of the restriction of $\tilde{f}$ to S^k which is given as part of non-uniform advice. In the i^{th} stage, table $T_i = \tilde{f}(a_1, \ldots, a_i, S^{k-i})$ is reconstructed. This reconstruction procedure uses table T_{i-1}, the circuit C and the set of indices S as input. The procedure (RECONSTRUCT) is described more formally below.

RECONSTRUCT$((a_1, \ldots, a_k), T_{i-1}, C, S)$

1 Forall $u \in S^{k-i}$ do
2 Guess $v \in \mathbf{F}^{2l}$ `/* This is a guess for the vector` $(\tilde{f}(a_1, \ldots, a_{i-1}, j, u))_{j \in \mathbf{F}}$ `*/`
3 Verify the following
  a $v \in \mathcal{L}$ `/*`v `is the table of a polynomial of low degree */`
  b $C(v) = 0$ `/* By guessing a setting of the choice bits of` C `making the circuit evaluate to 0 on` v `*/`
  c $\pi_S(v) = \pi_S(\tilde{f}(a_1, \ldots, a_{i-1}, j, u))_{j \in \mathbf{F}}$ `/* By looking up the table` T_{i-1} `*/`
4 If all the verifications are correct, then include v_{a_i} as the value of $\tilde{f}(a_1, \ldots, a_i, u)$ in T_i

Figure 1: Reconstructing T_i

Let v be a vector satisfying the conditions in Step 3 of the procedure. The verifications done in (a) and (b) guarantee that v is a vector in $\mathcal{L} \cap Z$. By Lemma 14, for all vectors $v' \in \mathcal{L} \cap Z$ other than v, their projection to the set of indices S is different. That is, $\pi_S(v) \neq \pi_S(v')$. Also, by the construction of the hitting set, $(\tilde{f}(a_1, \ldots, a_{i-1}, j, u))_{j \in \mathbf{F}} \in \mathcal{L} \cap Z$. Hence verification done in (c) ensures that $v = (\tilde{f}(a_1, \ldots, a_{i-1}, j, u))_{j \in \mathbf{F}}$ and hence $v_{a_i} = \tilde{f}(a_1, \ldots, a_i, u)$.

Let us estimate the size of the circuit for $\tilde{f}$ that can be built from the above procedure. In order for computing $\tilde{f}(a_i, \ldots, a_k)$, the procedure is called k times. For each such call, the time complexity of the above procedure is bounded by the time required to do less than $|S|^{k-1}$ verifications of a v-value. Each of these verifications takes the time of evaluating a circuit of size n^q, the time required to check that a table of size n is a low degree polynomial (which is bounded by n^2) and comparing $|S|$ values in F (which is bounded by n). Thus, building in the advice, we can convert the overall procedure into an SV-nondeterministic circuit. The size of the circuit is upper bounded by $O((n^\delta)^k n^q)$.

Now we can choose δ and k such that

$$O((n^\delta)^k n^q) < 2^{\epsilon k l}$$

It is very easily seen that choosing $\delta = \frac{\epsilon}{4}$ allows us to choose a constant $k = \frac{12q}{\epsilon}$, so that the above inequality is satisfied. ∎

Remark: It is not essential to use the low degree extension for our construction. What we essentially use is the fact that the error-correcting code we get from low degree multivariate polynomials is the *tensor product* of the low degree univariate polynomial codes. Therefore in our construction we can use the tensor product of any systematic code (codes which have the original message as part of the codeword) with parameters comparable to the low degree polynomial codes. Please see the survey paper [Mil01] for a construction based on tensor products of codes.

We can combine the above theorem with Lemma 13 to get Corollary 8.

Corollary 8 *For any constant $\epsilon > 0$, there is a constant $\tau > 0$ so that the following holds. There is a deterministic polynomial time procedure which, given as input the truth table of a Boolean function $f : \{0,1\}^m \to \{0,1\}$ (i.e., 2^m bits) with SV-nondeterministic circuit complexity at least $2^{\epsilon m}$, outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n, where $n = \lceil 2^{\tau m} \rceil$.*

Proof Let $\epsilon > 0$ be given. Assume m is sufficiently large. Given a truth table on m inputs with SV-nondeterministic circuit complexity at least $2^{\epsilon m}$, pad this truth table with zeros to obtain a truth table on m' inputs so that m' is divisible by k. Since k is a constant (to be chosen later), m' is at most $2m$ for sufficiently large m. Hence the circuit complexity of the new truth table is at least $2^{(\epsilon/2)m'}$.

Applying Theorem 7 (with parameters $\delta = \frac{\epsilon}{8}$, and $k = \frac{24q}{\epsilon}$), we have an efficient procedure for transforming this truth table into a hitting set in $\{0,1\}^{n'}$ with threshold $1 - 2^{-n' + n'^\delta}$ for co-nondeterministic circuits of size $(n')^q$, where $n' = (2m'/k)2^{2m'/k}$.

For the chosen value of δ, let γ be the constant in Lemma 13. Now choose $\tau = \gamma/k$ and apply Lemma 13 and deterministically convert this hitting set into a hitting set in $\{0,1\}^{n''}$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n'' with

$n'' = \lceil ((2m'/k)2^{2m'/k})^\gamma \rceil \geq 2^{2m\gamma/k} \geq \lceil 2^{\tau m} \rceil$.

Take this hitting set and remove the last $n'' - n$ bits in each string in it. This is the desired hitting set in $\{0,1\}^n$. ∎

Implications

Corollary 15 *For any constant $\epsilon > 0$, the following holds. If there exists a language L in* $\mathbf{NE} \cap \mathbf{coNE}$ *so that for all but finitely many n, $L \cap \{0,1\}^n$ requires SV-nondeterministic circuits of size $2^{\epsilon n}$, then there is an SVNP-procedure which on input 1^n generates a hitting set $H \subseteq \{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n.*

Proof Given ϵ, let τ be the corresponding constant of Corollary 8. On input 1^n, with n sufficiently large, the SVNP-procedure computes $m = \lceil \tau^{-1} \log n \rceil$ and enumerates the truth table of the characteristic function of L on $\{0,1\}^m$. Having found the truth table, it applies the procedure of Corollary 8 to it, yielding a hitting set in $\{0,1\}^{n'}$, where $n' = \lceil 2^{\tau m} \rceil = \lceil 2^{\tau \lceil \tau^{-1} \log n \rceil} \rceil$. Take this hitting set and remove the last $n' - n$ bits in each string in it. This is the desired hitting set in $\{0,1\}^n$. ∎

From Proposition 11 and Corollary 15 we have the derandomization result for **AM**.

Theorem 5 *Let $\epsilon > 0$ be any constant. If there exists a language L in* $\mathbf{NE} \cap \mathbf{coNE}$ *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{\epsilon n}$, then* **AM**=**NP**.

As graph non-isomorphism is in **AM** [GMW91] (and trivially in **coNP**), we have in particular the following corollary.

Corollary 16 *If for some $\epsilon > 0$, there exists a language $L \in \mathbf{NE} \cap \mathbf{coNE}$ so that for all but finitely many n, $L \cap \{0,1\}^n$ requires SV-nondeterministic circuits of size $2^{\epsilon n}$, then Graph Isomorphism is in* $\mathbf{NP} \cap \mathbf{coNP}$.

Corollary 8 also implies a derandomization of the class **BPP**, stated in Theorem 6. It is easy to see that, by a proof completely analogous to the proof of Corollary 15, we can get the following corollary.

Corollary 17 *For any constant $\epsilon > 0$, the following holds. If there is a language L in* **E** *so that the SV-nondeterministic circuit complexity of $L \cap \{0,1\}^n$ is at least $2^{\epsilon n}$ for all but finitely many n, then there is a polynomial time procedure which on input 1^n generates a hitting set $H \subseteq \{0,1\}^n$ with threshold $\frac{1}{2}$ for circuits of size n.*

The following result of [ACR98] (simplified by [ACRT97, BF99]) gives a method to derandomize **BPP** using hitting sets.

Theorem 18 (Andreev-Clementi-Rolim) *If there is a polynomial time procedure which on input 1^n outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for deterministic circuits of size n then* **BPP**=**P**.

Hence we obtain Theorem 6.

4 Relativizable hitting set generators are dispersers

The explicit construction of disperser with necessary parameters is a major technical tool that we employ in constructing our hitting set. In this section, we note that any *similar* construction of hitting sets has to appeal to the existence of explicit dispersers or itself provide such dispersers. More precisely, we note that any *relativizable, hardness-based* hitting sets are also dispersers with matching parameters.

We make it more formal for certain setting of parameter. Note that Corollary 8 and its proof relativize. i.e., the following statement has been proved.

Corollary 19 *For any $\epsilon > 0$, there is a $\delta > 0$ so that the following holds. There is a deterministic polynomial time procedure which, for any oracle A, has the following property. Given as input the truth table of a Boolean function $f : \{0,1\}^m \to \{0,1\}$ (i.e., 2^m bits) with SV-nondeterministic oracle circuit complexity with oracle gates for A at least $2^{\epsilon m}$, outputs a hitting set in $\{0,1\}^n$ with threshold $\frac{1}{2}$ for co-nondeterministic oracle circuits of size n with oracle gates for A, where $n = \lceil 2^{\delta m} \rceil$.*

The hardest part of a self-contained proof of Corollary 19 is the existence of explicit dispersers. We now note that any proof of Corollary 19 *has* to appeal to the existence of explicit dispersers or itself provide such dispersers.

Theorem 20 *Let a procedure with the property of Corollary 19 be given. Let $n = 2^m$ be sufficiently large. Define the bipartite graph $G_n = (U_n, V_n, E_n)$ with $U_n = \{0,1\}^n$, $V_n = \{0,1\}^{\lceil n^\delta \rceil}$, and an edge between x and y if and only if y is a member of the hitting set produced by the procedure on input x. Then G_n is a disperser with threshold $2^{n^{2\epsilon}}$.*

Proof We need to prove that, given any subset S of U_n with $|S| \geq 2^{n^{2\epsilon}}$, more than half the vertices of V_n are adjacent to S. Suppose not. Let S be a set for which this is not the case, and let A be the non-neighbours of S, so we have $|A| \geq |V_n|/2$. Viewed as a subset of $\{0,1\}^{\lceil n^\delta \rceil}$, we can use A as an oracle and consider circuit complexity relative to A. By Shannon's counting argument, viewed as truth tables for Boolean functions on n variables, at least one of the members of S must have SV-nondeterministic oracle circuit complexity with oracle gates for A at least $\frac{1}{2} \log |S| / \log \log |S| > n^\epsilon = 2^{\epsilon m}$. Let this element of S be denoted a. Thus, as the procedure has the property of Corollary 19, the vertices in V_n adjacent to a will intersect every set in V_n which

1. is the characteristic (accepted) set of an oracle circuit with oracle gates for A of size at most n and
2. has size at least $|V_n|/2$.

But then consider the oracle circuit defined by $x \to A(x)$. It has size n, its characteristic set has size at least $|V_n|/2$, and the neighbours of a do not intersect its characteristic set, as this set is the non-neighbours of S and $a \in S$. A contradiction. ■

5 Simulating AM in nondeterministic quasi-polynomial time

In this section we sketch how the techniques used to prove Theorem 5 can be directly applied to get other hardness-randomness tradeoffs for the class **AM**. In particular we show that if there are languages in **NEXP** $\cap$ **coNEXP** that require SV-nondeterministic circuit complexity $2^{n^{\frac{1}{2}+\epsilon}}$ for some $\epsilon > 0$, then **AM** is in nondeterministic quasi-polynomial time.

Let **NQuasiP** denote the class of problems that can be decided in nondeterministic quasi-polynomial time. That is, $L \in \mathbf{NQuasiP}$ if there exists a nondeterministic machine accepting L which runs in time $2^{\log^c n}$ for some constant c. By SVNQuasiP-procedure we mean a single valued nondeterministic procedure (defined formally in Section 2) running in time $2^{\log^c n}$ for some constant c.

We will show the following derandomization of **AM**.

Theorem 9 *Let $\epsilon > 0$ be any constant. If there exists a language L in* **NEXP**$\cap$**coNEXP** *so that for all but finitely many n, $L \cap \{0,1\}^n$ has SV-nondeterministic circuit complexity at least $2^{n^{\frac{1}{2}+\epsilon}}$, then* $\mathbf{AM} \subseteq \mathbf{NQuasiP}$.

As in the proof of Theorem 5, we first construct a hitting set with very high threshold, and convert this into a hitting set of threshold $\frac{1}{2}$ using explicitly constructed dispersers. The main difference from the proof of Theorem 5 is the choice of parameters for the dispersers. Here we need dispersers with much smaller threshold than that are used in proving Lemma 13. In particular, we need for any δ, explicit dispersers with threshold $2^{2^{\log^\delta n}}$. Since the proofs of most of the results required to prove Theorem 9 are very similar to the proofs of analogous statements that are given in previous sections, we only sketch them here.

Analogous to Proposition 11, we have the following proposition.

Proposition 21 *If there is an* SVNQuasiP*-procedure which on input 1^n outputs a hitting set in $\{0,1\}^n$ for co-nondeterministic circuits with threshold $\frac{1}{2}$, then* $\mathbf{AM} \subseteq \mathbf{NQuasiP}$.

An explicit construction of dispersers with parameters necessary for us is given in [TS02].

Theorem 22 (Ta-Shma) *For any constant $\delta > 0$ there is a $\gamma > 0$ such that an explicit $(2^{\log^\delta n}, 2^{\log^\gamma n})$-disperser exists.*

Using the disperser with the above-mentioned parameters, it is easy to rework the proof of Lemma 12 to prove the following.

Lemma 23 *For any constant $\delta > 0$, there are constants $\gamma > 0$ and $q \geq 1$ such that the following holds. There is a polynomial time procedure which takes input H, where $H \subseteq \{0,1\}^n$ is a hitting set for co-nondeterministic circuits of size n^q with threshold $1-2^{-n+2^{\log^\delta n}}$ and outputs a hitting set $H' \subseteq \{0,1\}^{n'}$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n', where $n' = \lceil 2^{\log^\gamma n} \rceil$.*

Now we state the main theorem of this section which is analogous to Theorem 7.

Theorem 24 *For any $\epsilon > \frac{1}{2}$, and $q \geq 1$, there is a polynomial time procedure P with the following properties. Let $f : \{0,1\}^m \to \{0,1\}$ be a function that cannot be computed by SV-nondeterministic circuits of size less than 2^{m^ϵ} for almost every m. Then there is a constant $\delta = \delta(\epsilon) < \epsilon$ and a $k = k(l)$ such that, given the truth-table of $f : \{0,1\}^{kl} \to \{0,1\}$ as input, P outputs a hitting set $H_f \subseteq \{0,1\}^n$ for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+2^{\log^\delta n}}$, where $n = (2l)2^{2l}$.*

Proof Sketch The construction of the hitting set is identical to the construction given in Theorem 7. We have to choose k and δ appropriately. Now suppose the set H_f produced is not a hitting set for co-nondeterministic circuits of size n^q with threshold $1 - 2^{-n+2^{\log^\delta n}}$. Then as in the proof of Theorem 7, we can construct an SV-nondeterministic circuit for f where the size of the circuit is $O((2^{\log^\delta n})^k(n^q))$. We will now chose δ and k such that $O((2^{\log^\delta n})^k(n^q)) < 2^{(kl)^\epsilon}$. An easy calculation shows that for large enough l, choosing $\delta = \frac{1}{2}(\epsilon - \frac{1}{2})$ allows us to choose $k = l$ so as to satisfy the inequality. ■

The above results implies Theorem 9. We sketch the proof here.

Proof Sketch (*of Theorem 9*) We give an SVNQuasiP-procedure (say M) which, on input 1^m outputs a hitting set in $\{0,1\}^m$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size m. Let f be the characteristic function of L.

For ϵ, Let δ be the constant given by Theorem 24. For this δ, let γ be the constant given by Lemma 23. Let $l = \lceil \log^{\frac{1}{\gamma}} m \rceil$.

The SVQNP-procedure, on input 1^m first enumerates the truth-table of f on inputs of length l^2. Now M, viewing this as a truth-table of $f : \{0,1\}^{kl} \to \{0,1\}$ (with $k = l$), simulates the deterministic procedure of Theorem 24 and produces a hitting set H_f in $\{0,1\}^{n'}$ for co-deterministic circuits of size $(n')^q$ with threshold $1 - 2^{-n'+2^{\log^\delta n'}}$, where $n' = (2l)2^{2l}$. Then M deterministically converts this hitting set H_f into a hitting set $H \subseteq \{0,1\}^{n''}$ with threshold $\frac{1}{2}$ for co-nondeterministic circuits of size n'', where $n'' = \lceil 2^{\log^\gamma n'} \rceil \geq m$. Finally, M removes the last $n'' - m$ bits from each string in H to produce the required hitting set. Since $l^2 = \log^{O(1)} m$, and $L \in \mathbf{NEXP} \cap \mathbf{coNEXP}$, it follows that M is an SVQNP-procedure. ■

6 Final Remarks

In addition to the derandomization of **AM**, Klivans and van Melkebeek [KvM02] gave several other applications of the fact that the Impagliazzo-Wigderson construction relativizes. Each of the applications showed that a hardness assumption involving oracle circuits implies a "derandomization" (in a loose sense).

For one of these extra applications we can combine their reasoning with Corollary 8 and obtain an improvement. Specifically, we can prove the following theorem relating two circuit lower bounds, which is identical to Theorem 5.15, [KvM02], except that there, “SV-nondeterministic circuit complexity” is replaced with “oracle circuit complexity with oracle gates for SAT”.

Theorem 25 *If there is a language L in $\mathbf{E}$ so that L has SV-nondeterministic circuit complexity at least $2^{\Omega(n)}$, then there exists a polynomially bounded function $p(n)$ and a polynomial-time computable family of matrices M_n where M_n is an $n \times n$ matrix over $\mathbf{Z}_{p(n)}[x]$ such that the linear transformation defined by the family M_n cannot be computed by log-depth linear size circuits which have special gates that can compute binary linear operators over $\mathbf{Z}_{p(n)}[x]$.*

We omit the proof which is a straightforward combination of the proof of Theorem 5.15 in Klivans and van Melkebeek and Corollary 8 of the present paper.

7 Recent Progress

Techniques in this paper work for functions in $\mathbf{NEXP} \cap \mathbf{coNEXP}$ with SV-nondeterministic circuit complexity at least 2^{n^δ} where $\delta > \frac{1}{2}$. Derandomizing $\mathbf{AM}$ under weaker hardness assumptions was open. Shaltiel and Umans [SU01] gave a construction which works for all ranges of parameters. In particular, they showed the “low-end” derandomization of $\mathbf{AM}$: under the assumption that $\mathbf{NEXP} \cap \mathbf{coNEXP}$ has functions with super-polynomial SV-nondeterministic circuit complexity, $\mathbf{AM} \subseteq \mathbf{NSUBEXP}$. Later, Umans gave optimal constructions for all ranges of hardness parameters [Uma03]. It is interesting to note that, like our construction, the constructions given in [SU01] and [Uma03] do not use *Nisan-Wigderson designs* which has been an important ingredient in most of the earlier derandomization results.

This paper uses non-uniform assumptions for derandomizing $\mathbf{AM}$. Many derandomization results based on uniform assumptions have been obtained recently. Lu, using the terminology of pseudo-classes introduced by Kabanets [Kab01], established a certain kind of uniform derandomization for $\mathbf{AM}$ [Lu01]. Impagliazzo, Kabanets, and Wigderson, under the assumption that $\mathbf{NEXP} \neq \mathbf{EXP}$, showed that $\mathbf{AM}$ can be simulated in nondeterministic sub-exponential time with sub-linear advice [IKW02]. Gutfreund, Shaltiel, and Ta-Shma observed certain resilience property of our hitting set construction to prove a uniform “high-end” derandomization result for $\mathbf{AM}$ [GSTS03].

There are essentially three circuit models that have been used in the literature for derandomizing $\mathbf{AM}$ under non-uniform assumptions: SAT-oracle circuits [KvM02], nondeterministic circuits [AK01, SU01], and SV-nondeterministic circuits (used in this paper). It is clear from the definitions (given in Section 2) that a SV-nondeterministic circuit of size s that computes a function f can be easily converted into a nondeterministic circuit of size $O(s)$ that computes the same function. On the other hand if complexity classes such as $\mathbf{NE} \cap \mathbf{coNE}$ or $\mathbf{E}$ which are closed under complement have nondeterministic circuits of size s, then they also have SV-nondeterministic circuits of size $O(s)$. A nondeterministic circuit can easily be seen as a restriction of a SAT-oracle circuit.

From the definitions it appears that the hardness assumption against SV-nondeterministic circuits is weaker than the corresponding hardness assumption against SAT-oracle circuits. A very recent work by Umans and Shaltiel [SU04] showed that these hardness assumptions are essentially equivalent. In particular, they established a surprising result that if $\mathbf{NE} \cap \mathbf{coNE}$ has oracle circuits of size s which make *nonadaptive* oracle queries to SAT then $\mathbf{NE} \cap \mathbf{coNE}$, also has SV-nondeterministic circuits of size $s^{O(1)}$. Since the proof of Theorem 4 given in [KvM02] goes through even under the assumption that there is a function in $\mathbf{NE} \cap \mathbf{coNE}$ that is hard against SAT-oracle circuits which make only nonadaptive queries, this result implies that the hardness assumption used in [KvM02] and the one used in this paper are essentially equivalent.

Finally, it is worth mentioning that in addition to results in [GSTS03] which use our construction, Barak et al. applied our main result to obtain certain results in cryptography [BOV03].

8 Acknowledgement

Both authors were supported by the ESPRIT Long Term Research Programme of the EU under project number 20244 (ALCOM-IT) and by BRICS, Basic Research in Computer Science, Centre of the Danish National Research Foundation.

We would like to thank Dieter van Melkebeek and Luca Trevisan for very helpful discussions.

References

[ACR96a] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Hitting properties of hard boolean operators and their consequences on BPP. Technical Report TR96-055, *Electronic Colloquium on Computational Complexity*, 1996. Available at `http://www.eccc.uni-trier.de/eccc`.

[ACR96b] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Hitting sets derandomize BPP. In *Proceedings of the 23rd International Colloquium on Automata, Languages and Programming*, volume 1099 of *Lecture Notes in Computer Science*, pages 357–368. Springer-Verlag, 1996.

[ACR97] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Worst-case hardness suffices for derandomization: A new method for hardness-randomness trade-offs. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, volume 1256 of *Lecture Notes in Computer Science*, pages 177–187. Springer-Verlag, 1997.

[ACR98] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. A new general derandomization method. *Journal of the ACM*, 45(1):179–213, January 1998.

[ACRT97] Alexander E. Andreev, Andrea E. F. Clementi, José D. P. Rolim, and Luca Trevisan. Weak random sources, hitting sets, and BPP simulations. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 264–272, 1997.

[AK01] V. Arvind and Johannes Köbler. On pseudorandomness and resource-bounded measure. *Theoretical Computer Science*, 255(1–2):205–221, 2001.

[Bab85] László Babai. Trading group theory for randomness. In *Proceedings of the 17th ACM Symposium on Theory of Computing*, pages 421–429, 1985.

[Bab92] László Babai. Bounded round interactive proofs in finite groups. *SIAM Journal on Discrete Mathematics*, 5(1):88–111, February 1992.

[BDG95] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity* II. Number 22 in EATCS Monographs on Theoretical Computer Science. Springer, Berlin-Heidelberg-New York, 1995.

[BF99] Harry Buhrman and Lance Fortnow. One-sided versus two-sided error in probabilistic computation. In *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science,*, volume 1563 of *Lecture Notes in Computer Science*, pages 100–109. Springer-Verlag, 1999.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd ACM Symposium on Theory of Computing*, pages 21–31, May 1991.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.

[BKL83] László Babai, William M. Kantor, and Eugene M. Luks. Computational complexity and the classification of finite simple groups. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, pages 162–171, 1983.

[BL83] László Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 171–183, 1983.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, November 1984.

[BM88] László Babai and Shlomo Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In *Proceedings of the 23rd International Cryptology Conference (CRYPTO 2003)*, volume 2729 of *Lecture Notes in Computer Science*, pages 229–315. Springer, 2003.

[FGM+89] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. In S. Micali, editor, *Advances in Computing Research 5: Randomness and Computation*, pages 429–442, 1989.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21th ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

[GS89] Shafi Goldwasser and Michael Sipser. *Private coins versus public coins in interactive proof systems*, volume 5 of *Advances in Computing Research*, pages 73–90. 1989.

[GSTS03] Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. Uniform hardness vs. randomness tradeoffs for Arthur-Merlin games. In *Proceedings of the 18th IEEE Conference of Computational Complexity*, pages 33–47, 2003.

[GVW00] Oded Goldreich, Salil Vadhan, and Avi Wigderson. Simplified derandomization of BPP using a hitting set generator. Technical Report TR00-004, *Electronic Colloquium on Computational Complexity*, 2000. Available at `http://www.eccc.uni-trier.de/eccc`.

[IKW02] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65:672–694, 2002.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pages 538–547, 1995.

[ISW99] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Near-optimal conversion of hardness into pseudo-randomness. In *Proceedings of the 40th IEEE symposium on Foundations of Computer Science*, pages 181–190, 1999.

[ISW00] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 1–10, 2000.

[IW97] Russell Impagliazzo and Avi Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[Kab01] Valentine Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. *Journal of Computer and System Sciences*, 63(2):236–252, 2001.

[KvM02] Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorhism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, Oct 2002.

[Lu01] Chi-Jen Lu. Derandomizing arthur-merlin games under uniform assumptions. *Computational Complexity*, 10:247–259, 2001.

[Mil98] Peter Bro Miltersen. Error correcting codes, perfect hashing circuits, and deterministic dynamic dictionaries. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 556–563, 1998.

[Mil01] Peter Bro Miltersen. Derandomizing complexity classes. *Handbook of Randomized Computing*, 2001. Kluwer Academic Publishers.

[MV99] Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing Arthur-Merlin games using hitting sets. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 71–80, 1999.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, October 1994.

[Pap94] Christopher Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Sel96] Alen L. Selman. Much ado about functions. In *Proceedings of the 11th IEEE Conference on Computational Complexity*, pages 198–212, 1996.

[Sha02] Ronen Shaltiel. Recent developments in extractors. *Bulletin of the European Association for Theoretical Computer Science*, 77:67–95, June 2002.

[Sip88] Michael Sipser. Expanders, randomness, or time versus space. *Journal of Computer and System Sciences*, 36:379–383, 1988.

[SSZ98] Michael Saks, Aravind Srinivasan, and Shiyu Zhou. Explicit OR-dispersers with polylogarithmic degree. *Journal of the ACM*, 45(1):123–154, Jan 1998.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[SU01] Ronen Shaltiel and Christopher Umans. Simple extractors for all minentropies and a new pseudo-random generator. In *Proceeding of the 41st IEEE Symposium on Foundations of Computer Science*, pages 648–657, 2001.

[SU04] Ronen Shaltiel and Christopher Umans. Pseudorandomness for approximate counting and sampling. *Not yet published*, 2004.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, July 2001.

[TS02] Amnon Ta-Shma. Almost optimal dispersers. *Combinatorica*, 22(1):123–145, 2002.

[TSUZ01] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Loss-less condensers, unbalanced expanders, and extractors. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 143–152, 2001.

[Uma03] Christopher Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*, 67:419–440, 2003.

[vM98] Dieter van Melkebeek. Derandomizing Arthur-Merlin games. Technical Report TR-98-08, The University of Chicago, Department of Computer Science, July 1998.

[Wil85] Chrisropher B. Wilson. Relativized circuit complexity. *Journal of Computer and System Sciences*, 31(2):169–181, 1985.

[Yao82] Andrew C. Yao. Theory and application of trapdoor functions. In *Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.