Weak Random Sources, Hitting Sets, and BPP Simulations

Alexander E. Andreev* Andrea E. F. Clementi† José D. P. Rolim‡ Luca Trevisan§

Abstract

We show how to simulate any BPP *algorithm in polynomial time using a weak random source of min-entropy r^γ for any $\gamma > 0$. This follows from a more general result about* sampling *with weak random sources. Our result matches an information-theoretic lower bound and solves a question that has been open for some years. The previous best results were a polynomial time simulation of* RP *[SSZ95] and a $n^{\log^{(k)} n}$-time simulation of* BPP *for fixed k [TS96].*

Departing significantly from previous related works, we do not use extractors; instead, we use the OR-disperser of [SSZ95] in combination with a tricky use of hitting sets borrowed from [ACR96].

Of independent interest is our new (simplified) proof of the main result of [ACR96]. Our proof also gives some new hardness/randomness trade-offs for parallel classes.

1 Introduction

Randomized algorithms are often the simpler ones to solve a given problem, or the most efficient, or both (see [MR95]). For some problems, including primality testing and approximation of # P-complete counting problems, only randomized solutions are known.

The practical applicability of such randomized methods depends on the effective possibility for an algorithm to access *truly random bits*. Since it is even questionable whether truly random sources really exist, much research has been devoted in the last decade to find weaker notions of randomness that are still sufficient to run BPP algorithms in polynomial time [VV85, SV86, Vaz86, Vaz87, CG88]. Several definitions of *weak random source* have been proposed in the literature, the most general being the following [CG88]: a source has min-entropy r^γ for $\gamma > 0$ if it outputs a string in $\{0,1\}^r$ and no string has probability of being output larger than 2^{r^γ}. An information-theoretic argument shows that a black-box simulation of BPP using a source with min-entropy $r^{o(1)}$ is impossible.

Dispersers and Extractors

The usual method to simulate a BPP algorithm using a weak random source is as follows. Say that, for a given input, the algorithm requires m (truly) random bits; then we ask the source r bits (note that it is required to make only one access to the weak random source since otherwise the problem would become trivial) and we use them to produce a sample space (a set of m-bit strings). Such strings are fed into the algorithm and then the majority rule is used to decide whether to accept or reject. The procedure that computes the sample space starting from the output of the source is *independent* of the algorithm that we want to derandomize. This simulation is basically equivalent [Zuc90, Zuc96b, NZ96, SZ94, SSZ95, TS96] to a bipartite graph $G = (V, W, E)$ having 2^r nodes in the left component V, 2^m nodes in the right component W, degree d and such that if we select a node v in the left component according to a weak random source of min-entropy r^γ, and then a random neighbour of v, the induced distribution in W is ϵ-close to the uniform distribution over W. Such graph is a $(2^r, 2^m, d, r^\gamma, \epsilon)$-*extractor*. The left nodes are seen as possible outcomes of the random source, the right nodes as possible random strings for the algorithm to be simulated. The simulation amounts to select a node in the left side according to the weak random source and then select as sample space the set of its neighbours. If, for some fixed γ, one could achieve d and r polynomial in m, then a polynomial time simulation of BPP would be possible using weak random sources of min-entropy r^γ. However, the best present construction of extractors for fixed $\gamma > 0$ and $r = \text{poly}(m)$ has $d = n^{log^{(k)} n}$ [TS96]. This implies a quasi-polynomial time simulation of BPP. A polynomial-time simulation of BPP using weak random sources of min-entropy r^γ for any fixed $\gamma > 0$ was one of the major open questions in the field.

It is not difficult to show that to simulate RP by means of a weak random source, *OR dispersers* [CW89] (from now on, we will simply call them *dispersers*) are sufficient. A $(2^r, 2^m, 2^{r^\gamma})$-disperser is again a bipartite graph $G = (V, W, E)$ with parameters r, m, and d as before, but now the property is that for any set $V' \subseteq V$ of at least 2^{r^γ} vertices on the left side and any set $W' \subseteq W$ of more than $2^m/2$ vertices on the right side, there is at least one edge joining V and W. This construction is somewhat easier to obtain, and Saks et al. [SSZ95] give indeed a disperser with $d = \text{poly}(n)$, for any constant $\gamma > 0$, allowing for a polynomial time simulation of RP.

See [Nis96] for a very complete and updated survey on extractors, dispersers, and weak random sources.

*andreev@mntn.msk.su. University of Moscow.

†clementi@dsi.uniroma1.it. University of Rome *La Sapienza*.

‡rolim@cui.unige.ch. University of Geneva.

§trevisan@cui.unige.ch. University of Geneva.

Pseudorandom Generators and Hitting Sets

A more ambitious goal than simulating BPP with weak random sources is the *deterministic* simulation of BPP. Research on this subject tries to isolate reasonable complexity assumptions under which deterministic simulations of randomized algorithms are possible [Nis90, NW94]. In some cases, combinatorial objects developed in the study of weak random sources have been used to give derandomization [NZ96]. Here we revert this connection, and we use a derandomization method to take full advantage from a weak random source.

Two basic combinatorial objects are studied in the theory of derandomization: pseudorandom generators (whose efficient construction immediately implies a deterministic simulation of BPP) and hitting set generators (whose efficient construction allows to simulate RP algorithms). Informally speaking, in the context of derandomization, pseudorandom generators play the same role of extractors and hitting sets generators play that of dispersers. A recent result of Andreev et al. [ACR96] shows how to deterministically simulate BPP algorithms using hitting set generators. This suggests that perhaps, dispersers could somewhat surprisingly be used to simulate BPP with weak random sources.

A quick ϵ-hitting set *generator* (ϵ-HSG) is an algorithm that, given a parameter n, finds in poly(n) time a set $H_n \subseteq \{0,1\}^*$ such that, for any finite Boolean function f of circuit complexity n, if $\mathbf{Pr}_{\vec{x}}[f(\vec{x}) = 1] \geq \epsilon$ then $f(\vec{a}) = 1$ for some $\vec{a} \in H_n$[1]. The probability is here computed with respect to the uniform distribution in $\{0,1\}^n$. The main result of [ACR96] can be formalized as follows.

Theorem 1 ([ACR96]) *For any choice of constants $\epsilon, \gamma > 0$, there is a deterministic algorithm that, given access to a quick γ-HSG, and given in input any circuit C of size n returns in* poly(n) *time a value A such that $|\mathbf{Pr}_{\vec{x}}[C(\vec{x}) = 1] - A| \leq \epsilon$. As a consequence, if for some $\gamma > 0$, a quick γ-HSG exists then* P = BPP.

The proof involves the construction of a set S of size poly(n) that is ϵ-discrepant for C, i.e. such that $\mathbf{Pr}_{\vec{x}\in S}[C(\vec{x}) = 1]$ approximates the value $\mathbf{Pr}_{\vec{x}}[C(\vec{x}) = 1]$ up to an additive error ϵ. A basic ingredient is the definition of a *discrepancy test* that enjoys a "soundness" property (Theorem 12): if the test accepts a set S, then S is ϵ-discrepant for C (the test requires a hitting set to meet this soundness property, but since we have a hitting set generator, this is not a problem). Thus, proving the theorem amounts to find a set S that passes the test. This task is solved in [ACR96] by means of a rather involved (and inherently sequential) algorithm that goes over $O(n \log n)$ local change phases of an initial set (that is the output of the hitting set generator) that is stored in a compressed form. The algorithm is of independent interest, since it proves a somewhat stronger result than Theorem 1 and has been also used in [ACR97]. For the sake of proving Theorem 1 it might however be over-kill.

Our Results

We show how to use *dispersers* and weak random sources to simulate BPP in polynomial time and to even solve a more general *sampling* problem. As an intermediate step of independent interest, we also present a new, simple, proof of Theorem 1.

In order to prove Theorem 1, we observe that the discrepancy test accepts often and that it has low circuit complexity, so that the hitting set generator itself can be used to generate a family of sets $S_1, \ldots, S_{k(n)}$ (where $k(n) = \text{poly}(n)$) one of whom is guaranteed to pass the test. Using this proof of Theorem 1 we note that the derandomized algorithm works in log-space (indeed, NC^1) when having oracle access to the γ-HSG and to an evaluation procedure for C. This has been used in [ACR97] to prove hardness/randomness trade-offs for parallel complexity classes.

The sampling problem we are interested in is as follows: given oracle access to a function $f : \{0,1\}^n \to \{0,1\}$ and a weak source of randomness, we want to solve the following task: find a set S of size poly(n) that with high probability is ϵ-discrepant for f. It should be clear that simulating a given BPP algorithm reduces to the above problem: the computation of a BPP algorithm on a fixed input is an (easy to compute) function f of the outcomes of the random coins. Being able to approximate the fraction of random coin outcomes that make f accept, allows to decide whether the algorithm accepts or not the input.

Whether this problem is solvable in polynomial time using a weak random source with min-entropy r^γ for any $\gamma > 0$ has been a long standing open question. Extractors were the only known tool to achieve results of this kind. We remark that extractors yield *oblivious sampling* algorithms (see [BR94, Zuc96a]).

We show that dispersers are sufficient. The starting point is the observation that using a disperser and a weak random source it is possible to generate polynomially many small sets $S_1, \ldots, S_k$ such that, with high probability, one of them is ϵ-discrepant for f, and sets $H_1, \ldots, H_k$ one of which (with high probability) has the hitting property required by the discrepancy test (see Theorem 19). Then, we define $H = \bigcup H_i$: since the hitting property is *monotone* (adding elements to a set cannot decrease its hitting properties), we have that H will be a hitting set with high probability. We can thus run the discrepancy test over the sets $S_1, \ldots, S_k$ using H as the reference hitting set. We shall then prove that, with high probability, one of the S_i's will pass the test and thus be ϵ-discrepant for f as required.

The main difference between our method and the extractor-based one is that the ϵ-discrepant set that is given in output *depends on* the specific function f that is accessed as oracle. The source of this non-obliviousness is the selection of a good set S_j among the candidates $S_1, \ldots, S_k$. As a result, our sampling algorithm is not *oblivious* according to the definition

[1] In the next section we will give a seemingly weaker (but in fact equivalent) formal definition.

of Bellare and Rompel [BR94], however it is *non-adaptive.*

Our main result can be stated in the following way

Theorem 2 *For any $\gamma > 0$, $\epsilon(n) = 1/\mathrm{poly}(n)$, there exists an algorithm that given a weak random source of min-entropy r^γ and oracle access to a function $f : \{0,1\}^n \to \{0,1\}$ computes, in time polynomial in n, a value A such that with probability at least $1-2^{-\mathrm{poly}(n)}$,*

$$|\mathbf{Pr}_{\vec{x}}[f(x) = 1] - A| \leq \epsilon .$$

Note that, since the algorithm runs in polynomial time, it will make poly(n) queries to f and, further, it will require only one access to the weak random source asking a bit sequence of poly(n) size.

Corollary 3 *For any $\gamma > 0$, any* BPP *algorithm can be simulated in polynomial time using a weak random source of min-entropy r^γ.*

Overview of the Paper

We give some definitions in Section 2. In Section 3 we present our new proof of Theorem 1. In Section 4 we show how to use a weak random source to generate with high probability a hitting set and a family of sets one of which is discrepant and we prove Theorem 2.

2 Preliminaries

Unless otherwise stated, probabilities are with respect to the uniform distribution. For any positive integer n we denote by $\mathcal{F}_n$ the set of all n-ary boolean functions $f : \{0,1\}^n \to \{0,1\}$. For a vector $\vec{a} \in \{0,1\}^n$, and a function $f : \{0,1\}^n \to \{0,1\}$, let $f^{\oplus \vec{a}}$ be defined as $f^{\oplus \vec{a}}(\vec{x}) = f(\vec{x} \oplus \vec{a})$.

We say that a Boolean function f *accepts* its input $\vec{x}$ if $f(\vec{x}) = 1$.

Definition 4 (Weak random source) *A probability distribution D over the set $\{0,1\}^r$ is a* weak random source of min entropy r^γ *if for any $x \in X$, $D(x) \leq 2^{-r^\gamma}$.*

For a vertex v of a graph $G = (V, E)$ we let $\Gamma(v) \subseteq V$ be the set of vertices that are adjacent to v. For a subset $S \subseteq V$, we define $\Gamma(S) = \bigcup_{v \in S} \Gamma(v)$. We give here a definition of dispersers which is more convenient than that given in the Introduction to describe our results. However, it is easy to check that they are equivalent.

Definition 5 (Disperser) *A bipartite multigraph $G(V, W, E)$ with $|V| = R$ and $|W| = N$ is said to be an (R, N, T)-disperser if for any subset $S \subseteq V$ such that $|S| \geq T$, it holds $\Gamma(S) \geq N/2$.*

Definition 6 (Circuit complexity) *For a boolean function $f : \{0,1\}^n \to \{0,1\}$ we denote by $L(f)$ the minimum size of a circuit computing f (here, for* circuit *we mean* a circuit whose gates have fan-in at most 2 and arbitrary fan-out.*)*

Definition 7 (Kolmogorov Complexity) *Let us fix a universal Turing machine U with alphabet $\{0,1\}$ for programs allowing oracle queries. Given two Boolean functions $f : \{0,1\}^k \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$, we define the* conditional Kolmogorov complexity of g given f *$K_U(g|f)$ as the shortest program for U that evaluates g having oracle access to f.*

For example, $K_U(f|f) = O(1)$. As usual, if we fix another universal Turing machine U' it holds $K_{U'}(g|f) = K_U(g|f) + \Theta(1)$. We will usually omit the subscript. See e.g. [LV90] for an introduction to Kolmogorov complexity. In this paper we will only need to use the obvious fact that, for any fixed f, the number of functions g such that $K(g|f) \leq k$ is at most 2^k.

Definition 8 (Hitting set) *A (multi)set $H \subseteq \{0,1\}^n$ is said to be ϵ-hitting for a family of functions $\mathcal{G} \subseteq \mathcal{F}_n$ if for any $f \in \mathcal{G}$ with $\mathbf{Pr}_{\vec{x}}[f(\vec{x}) = 1] \geq \epsilon$ there exists $\vec{x} \in H$ such that $f(\vec{x}) = 1$.*

Definition 9 (Discrepancy set) *A (multi)set $S \subseteq \{0,1\}^n$ is said to be ϵ-discrepant for a family of functions $\mathcal{G} \subseteq \mathcal{F}_n$ if for any $f \in \mathcal{G}$,*

$$|\mathbf{Pr}_{\vec{x} \in S}[f(\vec{x}) = 1] - \mathbf{Pr}_{\vec{x}}[f(\vec{x}) = 1]| \leq \epsilon .$$

Note that if a set is ϵ-discrepant for a family $\mathcal{G}$ then it is also ϵ'-hitting for $\mathcal{G}$ for any $\epsilon' < \epsilon$, but the converse is not necessarily true.

The definition below is a slight variant of the definition of quick ϵ-HSG of price $O(\log n)$ given in [ACR96].

Definition 10 (Hitting Set Generator) *A quick ϵ-HSG is a polynomial-time algorithm $\mathcal{H}$ that, given in input a number n in unary, returns a multiset $\mathcal{H}(n) \subseteq \{0,1\}^n$ that is ϵ-hitting for the set $\{f : \{0,1\}^n \to \{0,1\} : L(f) \leq n\}$.*

It may seem awkward to restrict the above definition to functions having circuit complexity equal to the number of inputs. However any n-ary function of circuit complexity K can be seen as a K-ary function of circuit complexity K whose value is independent of $K - n$ of its inputs (this point of view does not change the fraction of satisfying inputs). As a consequence of this observation, the set $\mathcal{H}(n)$ returned by the HSG hits *any* function of circuit complexity at most n.

The definition above is independent of the choice of ϵ, as long as it is a constant.

Lemma 11 ([ACR96]) *For any two constants $\epsilon, \epsilon' > 0$, if there exists a quick ϵ-HSG, then there exists a quick ϵ'-HSG.*

3 A New Proof of Theorem 1

The aim of this section is to show that, if we have a Boolean function $f : \{0,1\}^n \to \{0,1\}$, we can use a quick HSG to generate a polynomial sized family of (multi)sets, one of which is a discrepancy set for f.

We will then show that the *discrepancy test* (that also makes use of the HSG) can be correctly used to select the good set for f. All the results of this section (but Lemma 15) will also be used in the next section in order to prove Theorem 2.

For starters, we present the discrepancy test and its properties. For any vector $\vec{a} \in \{0,1\}^n$, function $f : \{0,1\}^n \to \{0,1\}$, and set $S \subseteq \{0,1\}^n$, define

$$p(\vec{a}, f, S) = \mathbf{Pr}_{\vec{x} \in S}[f(\vec{x} \oplus \vec{a}) = 1] \ .$$

For any two subsets $S, H \subseteq \{0,1\}^n$, constant $\epsilon > 0$, and function $f : \{0,1\}^n \to \{0,1\}$, we define the discrepancy test introduced in [ACR96].

```
DISC-TEST(f, S, H, ε)
begin
  p_min := min{p(a, f, S) : a ∈ H ∪ {0}};
  p_max := max{p(a, f, S) : a ∈ H ∪ {0}};
  if p_max − p_min ≤ ε then return (1)
                       else return (0)
end
```

Theorem 12 (Soundness of DISC-TEST **[ACR96])** *Constants c_0 and c_1 exist such that, for any $\epsilon > 0$, integer n, function $f : \{0,1\}^n \to \{0,1\}$, sets $S, H \subseteq \{0,1\}^n$, if* DISC-TEST$(f, S, H, \epsilon) = 1$ *and one of the following conditions holds*

1. *H is ϵ-hitting for the set of functions g such that $L(g) \leq c_0|S|L(f)$;*
2. *H is ϵ-hitting for the set of functions g such that $K(g|f) \leq c_1|S|n$;*

then S is 2ϵ-discrepant for f.

The sufficiency of condition (1) is used in this section to prove Theorem 1, while condition (2) is exploited in next section in order to prove Theorem 2.

Theorem 12 is the core of the results of [ACR96]. Note that it says that a set H with a certain *one-sided* pseudorandom property (the hitting property) can be used to test S for a *two-sided* pseudorandom property (the discrepancy property). However H has to be hitting for *a whole set* of functions while S is tested for discrepancy *on f only*. So, roughly speaking, the theorem trades-off "globality" versus "two-sidedness". We present a proof of Theorem 12 in the Appendix.

What remains to be done to prove Theorem 1 is to use the quick HSG to create a set S that passes the test. As a matter of fact, we will use the quick HSG to generate polynomially many sets $S_1, \ldots, S_k$ such that at least one of them passes the test.

We first give a sufficient condition for the DISC-TEST to accept.

Theorem 13 (Completeness of DISC-TEST**)** *If S is $(\epsilon/2)$-discrepant for the set $\{f^{\oplus\vec{a}} : \vec{a} \in \{0,1\}^n\}$, then* DISC-TEST$(f, S, H, \epsilon) = 1$*, for any set H.*

PROOF: Let $\vec{a}_1$ (resp. $\vec{a}_2$) be a point where $p_{\min} = p(\vec{a}_1, f, S)$ (resp. $p_{\max} = p(\vec{a}_2, f, S)$).

$$\begin{aligned} p_{\min} &= \mathbf{Pr}_{\vec{x} \in S}[f^{\oplus\vec{a}_1}(x) = 1] \\ &\geq \mathbf{Pr}_{\vec{x}}[f^{\oplus\vec{a}_1}(x) = 1] - \epsilon/2 \\ &= \mathbf{Pr}_{\vec{x}}[f(x) = 1] - \epsilon/2 \ . \end{aligned}$$

Where the first inequality is due to the discrepancy property of S. Similarly, we can prove

$$p_{\max} \leq \mathbf{Pr}_{\vec{x}}[f(x) = 1] + \epsilon/2 \ ,$$

and thus $p_{\max} - p_{\min} \leq \epsilon$. □

Lemma 14 (DISC-TEST **accepts often)** *Let $f : \{0,1\}^n \to \{0,1\}$, and let $H \subseteq \{0,1\}^n$ and $\epsilon > 0$ be fixed. Define $t : \{0,1\}^{mn} \to \{0,1\}$ as*

$$t(\vec{x}_1, \ldots, \vec{x}_m) = \text{DISC-TEST}(f, \{\vec{x}_1, \ldots, \vec{x}_m\}, H, \epsilon)$$

Then a constant c_2 exists such that, for any $m \geq \frac{1}{\epsilon^2}c_2 n$, t accepts at least a fraction $3/4$ of its inputs.

PROOF: Observe that a set of m randomly chosen elements from $\{0,1\}^n$ is not $\epsilon/2$-discrepant for a given function with probability at most $2^{-\Omega(m\epsilon^2)}$ (this follows from the well-known sampling lemma). Since

$$|\{f^{\oplus\vec{a}} : \vec{a} \in H \cup \{\vec{0}\}\}| \leq 2^n \ ,$$

we have that the probability that a randomly chosen set of $m \geq \frac{1}{\epsilon^2}c_1 n$ elements of $\{0,1\}^n$ is not $\epsilon/2$-discrepant for $\{f^{\oplus\vec{a}} : \vec{a} \in H \cup \{\vec{0}\}\}$ is at most

$$2^n 2^{-\Omega(m\epsilon^2)} \leq 1/4$$

for sufficiently large c_1. □

Lemma 15 (DISC-TEST **is easy)** *Let $t : \{0,1\}^{mn} \to \{0,1\}$ be defined as in Lemma 14. Then*

$$L(t) \ \leq \ (n + L(f))(|H| + 1)m + c_3 m|H|,$$

for some constant $c_3 > 0$.

PROOF: To compute the outcome of DISC-TEST it is necessary to evaluate, for any $\vec{a} \in H \cup \{\vec{0}\}$ and any $\vec{x} \in S$, $f(\vec{x} \oplus \vec{a})$. It is clear that $f(\vec{x} \oplus \vec{a})$ can be computed by a circuit of size $n + L(f)$ once $\vec{x}$ and $\vec{a}$ are available. After all those computations are performed, a sum and some comparisons have to be performed, and this can be done with a linear-sized circuit. □

PROOF: [Of Theorem 1] From Lemma 11 we can assume that we have a quick $\epsilon/2$-HSG (which will also be a $3/4$-HSG). Given in input a circuit C of size n, we set $m = 4c_2 n/\epsilon^2$, where c_2 is the constant of Lemma 14. We first use the HSG to generate a set H that is $\epsilon/2$-hitting for all the functions g such that $L(g) \leq c_0 mn$, where c_0 is the constant of Theorem 12. Such set will have size poly(n). Observe that Theorem 12 now implies that any set $S \subseteq \{0,1\}^n$ such that $|S| \leq m$ and DISC-TEST$(C, S, H, \epsilon/2)$ accepts is ϵ-discrepant for C (here we identify C with the Boolean function that it computes).

We then use again the HSG to generate such a set S. More formally, we use the HSG to generate a set $H' \subseteq \{0,1\}^{mn}$ that is 3/4-hitting for the set of functions $g : \{0,1\}^{mn} \to \{0,1\}$ such that $L(g) \leq 2(|H|+1)mn + c_3 m|H|$ (c_3 being the constant of Lemma 15). Let us define $t : \{0,1\}^{mn} \to \{0,1\}$ as $t(\vec{x}_1, \ldots, \vec{x}_m) =$ DISC-TEST$(C, \{\vec{x}_1, \ldots, \vec{x}_m\}, H, \epsilon/2)$. From Lemma 15 we have that $L(t) \leq 2(|H|+1)mn + c_3 m|H|$, while from Lemma 14 we have that $\mathbf{Pr}_{\vec{x}_1,\ldots,\vec{x}_m}[t(\vec{x}_1, \ldots, \vec{x}_m)] \geq 3/4$. Thus H' contains an element $(\vec{y}_1, \ldots, \vec{y}_m)$ that makes t accept. The set $S = \{\vec{y}_1, \ldots, \vec{y}_m\}$ is, therefore, ϵ-discrepant for C. We return $A = \mathbf{Pr}_{\vec{y} \in S}[C(\vec{y}) = 1]$. □

4 Proof of Theorem 2

The starting point of our proof is the following easy observation: if we have a set $I \subseteq \{0,1\}^N$ such that $\mathbf{Pr}_{\vec{x}}[\vec{x} \in I] > 1/2$, then using a weak random source and the dispersers of [SSZ95], we can generate a polynomial-sized (in N) set of vectors $\vec{x}_1, \ldots, \vec{x}_k$ such that, with high probability, $\{\vec{x}_1, \ldots, \vec{x}_k\} \cap I \neq \emptyset$. This is formalized in Lemma 17 below.

A naive way of using this fact would be to take the set I as the family of ϵ-discrepant sets S for f of size m. For large enough m ($m = O(n/\epsilon^2)$ would suffice) the set I will be such that $\mathbf{Pr}_{S \subseteq \{0,1\}^m}[S \in I] > 1/2$ and so we can use the weak random source to generate a family of sets $S_1, \ldots, S_k$ such that, with high probability, one of them is ϵ-discrepant for f. But now the problem is that we are not able to recognize which of these sets has the discrepancy property (note that an efficient algorithm to test the discrepancy property would imply ZPP = BPP). Theorem 12 gives indeed a way to test for discrepancy, provided that we have a hitting set at hand.

We thus define $I \subseteq \{0,1\}^{n(m+M)}$ as the family of pairs of sets (H, S) such that H has M elements and the hitting property as in the hypothesis of Theorem 12 and S has m elements and the discrepancy property as in the hypothesis of Theorem 13. As shown in Lemma 18 below, for an appropriate choice of m and M, the set I is such that $\mathbf{Pr}_{(H,S)}[(H, S) \in I] > 1/2$. Using the weak random source we can thus obtain a set of pairs $(H_1, S_1), \ldots, (H_k, S_k)$ such that, for some j, S_j has the required discrepancy property and H_j the required hitting property (with high probability). The next important observation is that the hitting property is *monotone*, that is, if a set H has a certain hitting property, and J is any set, then $H \cup J$ has at least the same hitting property of H (the reader may want to verify that the discrepancy property is *not* monotone). As a consequence, the set $\bigcup_i H_i$ has (with high probability) the hitting property required by Theorem 12.

We start quoting the disperser construction of Saks, Srinivasan, and Zhou [SSZ95].

Theorem 16 (Construction of dispersers) *For any $0 < \lambda < \alpha \leq 1$, for any sufficiently large r, and for any $2^{r^\alpha} \leq T \leq 2^r$, there exists an efficient construction of an $(2^r, 2^{r^\lambda}, T)$-disperser $G = (V, W, E)$ with* poly(r) *degree.*

In the previous theorem, by "efficient construction" we mean the existence of an algorithm that for any vertex of V finds its neighbours in time poly(r).

Lemma 17 *For any fixed $\gamma > 0$, there exists a polynomial time algorithm that, on input $n > 0$, using a weak random source of min-entropy r^γ, outputs a set $S \subseteq \{0,1\}^n$ of size* poly(n) *such that, for any $I \subseteq \{0,1\}^n$ with $|I| > 2^{n-1}$, it holds*

$$\mathbf{Pr}[S \cap I = \emptyset] \leq 2^{-\mathrm{poly}(n)} .$$

PROOF: Fix λ and α such that $0 < \lambda < \alpha < \gamma$. Let $r = n^{1/\lambda}$. Consider a $(2^r, 2^n, 2^{r^\alpha})$-disperser $G = (V, W, E)$, that can be efficiently constructed as in Theorem 16. We identify V with $\{0,1\}^r$ and W with $\{0,1\}^n$. Let $B \subseteq V$ be the set of "bad" vertices v such that $\Gamma(v) \subseteq W - I$. We claim that $|B| < 2^{r^\alpha}$: indeed, otherwise, we would have $|\Gamma(B)| \geq 2^n/2$, while $|W - I| < 2^n/2$. Let us choose an element v of V by querying r bits from a weak random source of min-entropy r^γ, and let S be the set of its neighbours. The probability that we picked a bad vertex $v \in B$ is at most $2^{r^\alpha} \cdot 2^{-r^\gamma} \leq 2^{-\mathrm{poly}(n)}$. On the other hand, if $v \notin B$, then $S \cap I \neq \emptyset$; the lemma thus follows. □

Lemma 18 *Constants c_4 and c_5 exist such that for any $\epsilon > 0$, $n > 0$, $f : \{0,1\}^n \to \{0,1\}$, $c > 0$ and for $m = c_4 n/\epsilon^2$ and $M = c_5 cmn/\epsilon$, for a randomly chosen element $(\vec{v}, \vec{u})$ (where $\vec{v} \in \{0,1\}^{nM}$ and $\vec{u} \in \{0,1\}^{nm}$) the following holds with probability larger than 1/2:*

1. *$\vec{v}$, regarded as a multiset of $\{0,1\}^n$ of size M, is ϵ-hitting for the set of functions g such that $K(g|f) \leq cmn$.*
2. *$\vec{u}$, regarded as a multiset of $\{0,1\}^n$ of size m, form a set that is $\epsilon/2$-discrepant for $\{f^{\oplus \vec{a}} : \vec{a} \in \{0,1\}^n\}$.*

PROOF: It suffices to prove that each event holds with probability larger than 3/4.

Regarding the first event, the number of functions $g \in \mathcal{F}_n$ such that $K(g|f) \leq cmn$ is clearly at most 2^{cmn}. If one such g has $\mathbf{Pr}_{\vec{x}}[g(\vec{x}) = 1] \geq \epsilon]$, then the probability that M randomly chosen elements from $\{0,1\}^n$ do not hit g is at most

$$(1 - \epsilon)^M \leq e^{-\epsilon M} .$$

Since $M = c_5 cmn/\epsilon$, it follows that, for a proper choice of c_5, the probability that all the functions g are hit is at least

$$1 - 2^{cmn} e^{-\epsilon M} > 3/4 .$$

The second claim easily follows from the proof of Lemma 14. □

Theorem 19 *For all $\epsilon, \gamma, c > 0$, there exists a polynomial-time algorithm that, using a weak random source with min-entropy r^γ, and oracle access to a function $f \in \mathcal{F}_n$, computes sets $H, S_1, \ldots, S_k \subseteq \{0,1\}^n$ such that the following holds with probability at least $1 - 2^{-\mathrm{poly}(n)}$:*

1. *$|S_1| = |S_2| = \cdots = |S_k|$.*
2. *H is ϵ-hitting for the set of functions g such that $K(g|f) \le c|S_1|n$;*
3. *for some $j \in \{1, \ldots, k\}$, S_j is $\epsilon/2$-discrepant for the set of functions $\{f^{\oplus \vec{a}} : \vec{a} \in \{0,1\}^n\}$.*

PROOF: By combining Lemmas 17 and 18 we can use a weak random source to generate sets $S_1, \ldots, S_k$ and $H_1, \ldots, H_k$ such that, with probability at least $1-2^{-\mathrm{poly}(n)}$, for some j, S_j (resp. H_j) has the required discrepancy (resp. hitting) property. Since the hitting property is *monotone* (adding elements to a hitting set does not decrease its hitting property) we also have that, with the same probability, $H = \bigcup_j H_j$ is ϵ-hitting for the set of functions g with $K(g|f) \le c|S_1|n$. □

We are now ready to prove Theorem 2.

PROOF: [Of Theorem 2] We generate a set H and sets $S_1, \ldots, S_k$ as in Theorem 19. We then run DISC-TEST(f, S_i, H, ϵ) for $i = 1, \ldots, k$ and we return S_j where j is the smallest index such that DISC-TEST(f, S_j, H, ϵ) accepts. From Theorem 13 we have that, with probability at least $1 - 2^{-\mathrm{poly}(n)}$, at least one such index exists, and from Theorem 12 we have (with the same probability) that the selected set is 2ϵ-discrepant for f. We then set $A = \mathbf{Pr}_{\vec{x} \in S}[f(\vec{x}) = 1]$. □

5 Conclusions

We have demonstrated how to simulate BPP algorithms in polynomial time using weak random sources with the smallest possible min-entropy.

The main novelty in our result has been the use of *dispersers* in a context where *extractors* seemed to be necessary. Extractors have other applications besides the use of weak random sources (see e.g. [Nis96]). It could be case that techniques similar to ours can give stronger results or simplified proofs in these other applications as well.

We also emphasize that our simulation runs in NC and, furthermore, it is possible to give an NC construction of the SSZ-dispersers [SSZ97]. This implies that our method also provides an efficient simulation of BPNC algorithms using weak random sources.

Andreev et al. [ACR97] have recently used our NC proof of Theorem 1 in order to provide sufficient conditions (in terms of worst-case circuit complexity) for NC = BPNC. Without our result they were only able to provide sufficient conditions for ZPNC = BPNC.

Acknowledgments

We are grateful to Oded Goldreich for helpful comments and suggestions on a preliminary version of this paper. In particular, the use of a counting argument *á la* Kolmogorov is due to Oded. We thank Michael Saks, Aravind Srinivasan and Shiyu Zhou for showing us that OR-dispersers can be obtained by an NC construction, and for other helpful conversations.

References

[ACR96] A.E. Andreev, A.E.F. Clementi, and J.D.P. Rolim. Hitting sets derandomize BPP. In *Proceedings of the 23rd International Colloquium on Automata, Languages and Programming*, pages 357–368. LNCS 1099, Springer-Verlag, 1996.

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

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, pages 276–287, 1994.

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, April 1988.

[CW89] A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 14–19, 1989.

[LV90] M. Li and P. Vitany. Kolmogorov complexity and its applications. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A*, pages 187–254. Elsevier, 1990.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Nis90] N. Nisan. *Using Hard Problems to Create Pseudorandom Generators*. MIT Press, 1990. ACM Distinguished Dissertations.

[Nis96] N. Nisan. Extracting randomness: How and why. In *Proceedings of the 11th IEEE Conference on Computational Complexity*, pages 44–58, 1996.

[NW94] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994. Preliminary version in *Proc. of FOCS'88*.

[NZ96] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996. Preliminary version in *Proc. of STOC'93.*

[SSZ95] M. Saks, A. Srinivasan, and S. Zhou. Explicit dispersers with polylog degree. In *Proceedings of the 27th ACM Symposium on Theory of Computing*, pages 479–488, 1995.

[SSZ97] M. Saks, A. Srinivasan, and S. Zhou. Personal communication, March 1997.

[SV86] M. Santha and U. Vazirani. Generating quasi-random sequences from slightly random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[SZ94] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, pages 264–275, 1994.

[TS96] A. Ta-Shma. On extracting randomness from weak random sources. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 276–285, 1996.

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

[Vaz87] U. Vazirani. Efficiency considerations in using semi-random sources. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 160–168, 1987.

[VV85] U. Vazirani and V. Vazirani. Random polynomial time is equal to slightly random polynomial time. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 417–428, 1985.

[Zuc90] D. Zuckerman. General weak random sources. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 534–543, 1990.

[Zuc96a] D. Zuckerman. Randomness-optimal sampling, extractors and constructive leader election. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 286–295, 1996.

[Zuc96b] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, 1996. Preliminary version in *Proc. of FOCS'91.*

```
function BAD(a)
constants
  x[1] := s⃗_1;
  x[2] := s⃗_2;
  ...
  x[m] := s⃗_m;
begin
  count := 0;
  for i := 1 to m do
    count := count + f(a ⊕ x[i]);
  if count > m p_max or count < m p_min then
    return (1)
  else
    return (0)
end.
```

Figure 1: How to compute BAD.

A Appendix

A.1 Proof of Theorem 12

PROOF: Let f, $S = \{\vec{s}_1, \ldots, \vec{s}_m\}$, H, ϵ be fixed. Let us define the function BAD : $\{0,1\}^n \to \{0,1\}$ as follows

$$\textsc{bad}(\vec{a}) = \begin{cases} 0 & \text{if } p_{\min} \leq p(\vec{a}, f, S) \leq p_{\max}; \\ 1 & \text{otherwise.} \end{cases}$$

Claim 20 *$L(\textsc{bad}) = O(|S|(L(f) + n))$ and $K(\textsc{bad}|f) \leq mn + 2\log m + O(1)$.*

PROOF: [Of Claim 20] To compute BAD, we first evaluate f over all the points of the form $\vec{a} \oplus \vec{x}$ for $\vec{x} \in S$. This requires at most $|S|n$ gates to compute the evaluation points and $nL(f)$ gates to perform the evaluations. This gives $|S|p(\vec{a}, S)$ in unary; the comparison between this value and $|S|p_{\min}$ and $|S|p_{\max}$ can be done with a circuit of $O(|S|)$ size (note that $p_{\min}$ and $p_{\max}$ are fixed, since they only depend on f, S, and H).

Regarding the bound on $K(\textsc{bad}|f)$, we observe that BAD can be computed with the pseudo-code depicted in Figure 1 The size of the program is $mn + 2\log m + O(1)$, since the definitions of the constants require mn bits, and the rest of the program has constant size, besides the explicit values of $mp_{\max}$ and $mp_{\min}$ (that can be rounded to the closest integer) that require $\log m$ bits each. □

We assume that c_0 (respectively, c_1) is large enough so that H is ϵ-hitting for BAD.

Claim 21 *$\mathbf{Pr}_{\vec{a}\in\{0,1\}^n}[\textsc{bad}(\vec{a}) = 1] < \epsilon$.*

PROOF:[Of Claim 21] Assume, by contradiction, that $\mathbf{Pr}_{\vec{a}\in\{0,1\}^n}[\textsc{bad}(\vec{a}) = 1] \geq \epsilon$, then by the hitting property of H, there exists some $\vec{a} \in H$ such that $\textsc{bad}(\vec{a}) = 1$, which is impossible by definition of BAD. □

Let $\mathbf{E}[p(\vec{a}, f, S)]$ be the average of $p(\vec{a}, f, S)$ over the choices of $\vec{a}$.

Claim 22 $\mathbf{E}[p(\vec{x}, f, S)] = \mathbf{Pr}_{\vec{x}}[f(\vec{x}) = 1]$.

PROOF: [Of Claim 22]

$$\begin{aligned}
\mathbf{E}[p(\vec{x}, f, S)] &= \mathbf{Pr}_{\vec{x}\in\{0,1\}^n\ ,\ \vec{y}\in S}[f(\vec{x}\oplus\vec{y}) = 1] \\
&= \frac{1}{|S|}\sum_{\vec{y}\in S}\mathbf{Pr}_{\vec{x}}[f(\vec{x}\oplus\vec{y}) = 1] \\
&= \frac{1}{|S|}\sum_{\vec{y}\in S}\mathbf{Pr}_{\vec{x}}[f(\vec{x}) = 1] \\
&= \mathbf{Pr}_{\vec{x}}[f(\vec{x}) = 1]\ .
\end{aligned}$$

□

From Claim 21 we have the following inequalities

$$\begin{aligned}
&\mathbf{E}[p(\vec{a}, f, S)] \leq (1-\epsilon)p_{\max} + \epsilon\cdot 1 < p_{\max} + \epsilon \\
&\mathbf{E}[p(\vec{a}, f, S)] \geq (1-\epsilon)p_{\min} \geq p_{\min} - \epsilon
\end{aligned}$$

If the test accepts, then $p_{\max} - p_{\min} \leq \epsilon$, and $p_{\min} \leq \mathbf{Pr}_{\vec{x}\in S}[f(\vec{x}) = 1] \leq p_{\max}$. It follows that

$$|\mathbf{Pr}_{\vec{x}}[f(\vec{x}) = 1] - \mathbf{Pr}_{\vec{x}\in S}[f(\vec{x}) = 1]| \leq 2\epsilon\ .$$

□