Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 4 (1999)

Almost k-Wise Independence and Hard Boolean Functions

Valentine Kabanets
Department of Computer Science
University of Toronto
Toronto, Canada
kabanets@cs.toronto.edu

November 5, 1999

Abstract

Andreev et al. [ABCR97] gave constructions of Boolean functions (computable by polynomial-size circuits) with large lower bounds for read-once branching program (1-b.p.'s): a function in P with the lower bound $2^{n-\mathrm{polylog}(n)}$, a function in quasipolynomial time with the lower bound $2^{n-O(\log n)}$, and a function in LINSPACE with the lower bound $2^{n-\log n-O(1)}$. We point out alternative, much simpler constructions of such Boolean functions by applying the idea of almost k-wise independence more directly, without the use of discrepancy set generators for large affine subspaces; our constructions are obtained by derandomizing the probabilistic proofs of existence of the corresponding combinatorial objects. The simplicity of our new constructions also allows us to observe that there exists a Boolean function in $\mathrm{AC}^0[2]$ (computable by a depth 3, polynomial-size circuit over the basis $\{\wedge, \oplus, 1\}$) with the optimal lower bound $2^{n-\log n-O(1)}$ for 1-b.p.'s.

Keywords: almost k-wise independence, derandomization, exponential lower bounds for read-once branching programs, r-mixed Boolean functions.

1 Introduction

Branching programs represent a model of computation that measures the space complexity of Turing machines. Recall that a *branching program* is a directed acyclic graph with one source and with each node of out-degree at most 2. Each node of out-degree 2 (a branching node) is labeled by an index of an input bit, with one outgoing edge labeled by 0, and the other by 1; each node of out-degree 0 (a sink) is labeled by 0 or 1. The branching program accepts an input if there is a path from the source to a sink labeled by 1 such that, at each branching node of the path, the path contains the edge labeled by the input bit for the input index associated with that node. Finally, the *size* of a branching program is defined as the number of its nodes.

While there are no nontrivial lower bounds on the size of general branching programs, strong lower bounds were obtained for a number of explicit Boolean functions in restricted models (see, e.g., [Raz91] for a survey). In particular, for *read-once branching programs (1-b.p.'s)* — where, on every path from the source to a sink, no two branching nodes are labeled by the same input index — exponential lower bounds of the form $2^{\Omega(\sqrt{n})}$ were given for explicit n-variable Boolean functions in [Weg88, Zak84, Dun85, Juk88, KMW91, SS93, Pon99, Gal97, BW98] among others. Moreover, [Juk88, KMW91, Gal97, BW98] exhibited Boolean functions in AC^0 that require 1-b.p.'s of size at least $2^{\Omega(\sqrt{n})}$.

ISSN 1433-8092

After lower bounds of the form $2^{\Omega(\sqrt{n})}$ were obtained for 1-b.p.'s, the natural problem was to find an explicit Boolean function with the truly exponential lower bound $2^{\Omega(n)}$. The first such bound was proved in [ABH+86] for the Boolean function computing the parity of the number of triangles in a graph; the constant factor was later improved in [SS93]. With the objective to improve this lower bound, Savický and Žák [SZ96] constructed a Boolean function in P that requires a 1-b.p. of size at least $2^{n-3\sqrt{n}}$, and gave a probabilistic construction of a Boolean function requiring a 1-b.p. of size at least $2^{n-O(\log n)}$. Finally, Andreev et al. [ABCR97] presented a Boolean function in LINSPACE $\cap$ P/poly with the optimal lower bound $2^{n-\log n+O(1)}$, and, by derandomizing the probabilistic construction in [SZ96], a Boolean function in QP $\cap$ P/poly with the lower bound $2^{n-O(\log n)}$, as well as a Boolean function in P with the lower bound $2^{n-\mathrm{polylog}(n)}$; here QP stands for the quasipolynomial time $n^{\mathrm{polylog}(n)}$.

The combinatorics of 1-b.p.'s is quite well understood: a theorem of Simon and Szegedy [SS93], generalizing the ideas of many papers on the subject, provides a way of obtaining strong lower bounds. A particular case of this theorem states that any 1-b.p. computing an r-mixed Boolean function has size at least $2^r - 1$. Informally, an r-mixed function essentially depends on every set of r variables (see the next section for a precise definition). The reason why this lower-bound criterion works can be summarized as follows. A subprogram of a 1-b.p. G_n starting at a node v does not depend on any variable queried along any path going from the source s of G_n to v, and hence v completely determines a subfunction of the function computed by G_n. If G_n computes an r-mixed Boolean function f_n, then any two paths going from s to v can be shown to query the same variables, whenever v is sufficiently close to s. Hence, such paths must coincide, i.e., assign the same values to the queried variables; otherwise, two different assignments to a set of at most r variables yield the same subfunction of f_n, contradicting the fact that f_n is r-mixed. It follows that, near the source, G_n is a complete binary tree, and so it must have exponentially many nodes.

Andreev et al. [ABCR97] construct a Boolean function $f_n(x_1, \ldots, x_n)$ in LINSPACE $\cap$ P/poly that is r-mixed for $r = n - \lceil \log n \rceil - 2$ for almost all n. By the lower-bound criterion mentioned above, this yields the optimal lower bound $\Omega(2^n/n)$ for 1-b.p.'s. A Boolean function in DTIME$(2^{\log^2 n}) \cap$ P/poly that requires a 1-b.p. of size at least $2^{n-O(\log n)}$ is constructed by reducing the amount of randomness used in the probabilistic construction of [SZ96] to $O(\log^2 n)$ advice bits. Since these bits turn out to determine a polynomial-time computable function with the lower bound $2^{n-O(\log n)}$, one gets a function in P with the lower bound $2^{n-O(\log^2 n)}$ by making the advice bits a part of the input.

Both constructions in [ABCR97] use the idea of ϵ-biased sample spaces introduced by Naor and Naor [NN93], who also gave an algorithm for generating small sample spaces; three simpler constructions of such spaces were later given by Alon et al. [AGHP92]. Andreev et al. define certain ϵ-discrepancy sets for systems of linear equations over GF(2), and relate these discrepancy sets to the biased sample spaces of Naor and Naor through a reduction lemma. Using a particular construction of a biased sample space (the powering construction from [AGHP92]), Andreev et al. give an algorithm for generating ϵ-discrepancy sets, which is then used to derandomize both a probabilistic construction of an r-mixed Boolean function for $r = n - \lceil \log n \rceil - 2$ and the construction in [SZ96] mentioned above.

Our results. We will show that the known algorithms for generating small ϵ-biased sample spaces can be applied *directly* to get the r-mixed Boolean function as above, and to derandomize the construction in [SZ96]. The idea of our first construction is very simple: treat the elements (bit strings) of an ϵ-biased sample space as the truth tables of Boolean functions. This will induce a probability distribution on Boolean functions such that, on any subset A of k inputs, the restriction

to A of a Boolean function chosen according to this distribution will look almost as if it were a uniformly chosen random function defined on the set A. By an easy probabilistic argument, we will show that such a space of functions will contain the desired r-mixed function, for a suitable choice of parameters ϵ and k.

We indicate several ways of obtaining an r-mixed Boolean function with $r = n - \lceil \log n \rceil - 2$. In particular, using Razborov's construction of ϵ-biased sample spaces that are computable by $\mathrm{AC}^0[2]$ formulas [Raz88] (see also [Sav95]), we prove that there are such r-mixed functions that belong to the class of polynomial-size depth 3 formulas over the basis $\{\&, \oplus, 1\}$. This yields the smallest (nonuniform) complexity class known to contain Boolean functions with the optimal lower bounds for 1-b.p.'s. (We remark that, given our lack of strong circuit lower bounds, it is conceivable that the characteristic function of every language in EXP can be computed in nonuniform $\mathrm{AC}^0[6]$.)

In our second construction, we derandomize a probabilistic existence proof in [SZ96]. We proceed along the usual path of derandomizing probabilistic algorithms whose analysis depends only on almost k-wise independence rather than full independence of random bits [NN93]. Observing that the construction in [SZ96] is one such algorithm, we reduce its randomness complexity to $O(\log^3 n)$ bits (again treating strings of an appropriate sample space as truth tables). This gives us a $\mathrm{DTIME}(2^{O(\log^3 n)})$-computable Boolean function of quasilinear circuit-size with the lower bound for 1-b.p.'s slightly better than that for the corresponding quasipolynomial-time computable function in [ABCR97], and a Boolean function in quasilinear time, QL, with the lower bound for 1-b.p.'s at least $2^{n-O(\log^3 n)}$, which is only slightly worse than the lower bound for the corresponding polynomial-time function in [ABCR97]. In the analysis of our construction, we employ a combinatorial lemma due to Razborov [Raz88], which bounds from above the probability that none of n events occur, given that these events are almost k-wise independent.

The remainder of the paper. In the following section, we state the necessary definitions and some auxiliary lemmas. In Section 3, we show how to construct an r-mixed function that has the same optimal lower bound for 1-b.p. as that in [ABCR97], and observe that such a function can be computed in $\mathrm{AC}^0[2]$. In Section 4, we give a simple derandomization procedure for a construction in [SZ96], obtaining two more Boolean functions (computable in polynomial time and quasipolynomial time, respectively) that are hard with respect to 1-b.p.'s.

2 Preliminaries

Below we give the standard definitions of k-wise independence and (ϵ, k)-independence. We consider probability distributions that are uniform over some set $S \subseteq \{0,1\}^n$; such a set is denoted by S_n and called a *sample space*.

Let S_n be a sample space, and let $X = x_1 \ldots x_n$ be a string chosen uniformly from S_n. Then S_n is *k-wise independent* if, for any k indices $i_1 < i_2 < \cdots < i_k$ and any k-bit string α, we have $\mathbf{Pr}[x_{i_1} x_{i_2} \ldots x_{i_k} = \alpha] = 2^{-k}$. Similarly, for S_n and X as above, S_n is *(ϵ, k)-independent* if $|\mathbf{Pr}[x_{i_1} x_{i_2} \ldots x_{i_k} = \alpha] - 2^{-k}| \leqslant \epsilon$ for any k indices $i_1 < i_2 < \cdots < i_k$ and any k-bit string α.

Naor and Naor [NN93] present an efficient construction of small (ϵ, k)-independent sample spaces; three simpler constructions are given in [AGHP92]. Here we recall just one construction from [AGHP92], the powering construction, although any of their three constructions could be used for our purposes.

Consider the Galois field $\mathrm{GF}(2^m)$ and the associated m-dimensional vector space over $\mathrm{GF}(2)$. For every element u of $\mathrm{GF}(2^m)$, let $\mathrm{bin}(u)$ denote the corresponding binary vector in the associated

vector space. The sample space $\mathrm{Pow}_N^{2^m}$ is defined as a set of N-bit strings such that each string ω is determined as follows. Two elements $x, y \in \mathrm{GF}(2^m)$ are chosen uniformly at random. For each $1 \leqslant i \leqslant N$, the ith bit ω_i is defined as $\langle \mathrm{bin}(x^i), \mathrm{bin}(y) \rangle$, where $\langle a, b \rangle$ denotes the inner product over GF(2) of binary vectors a and b.

The next lemma follows from the results in [AGHP92] (Proposition 3 and Corollary 1).

Lemma 1 ([AGHP92]) *For every $k \leqslant N$, the sample space $\mathrm{Pow}_N^{2^m}$ is $(\frac{N}{2^m}, k)$-independent.*

As we have mentioned in the introduction, we shall view the strings of the sample space $\mathrm{Pow}_N^{2^m}$ as the truth tables of Boolean functions of $\log N$ variables. It will be convenient to assume that N is a power of 2, i.e., $N = 2^n$. Thus, the uniform distribution over the sample space $\mathrm{Pow}_{2^n}^{2^m}$ induces a distribution $\mathbf{F}_{n,m}$ on Boolean functions of n variables that satisfies the following lemma.

Lemma 2 *Let A be any set of k strings from $\{0,1\}^n$, for any $k \leqslant 2^n$. Let ϕ be any Boolean function defined on A. For a Boolean function f chosen according to the distribution $\mathbf{F}_{n,m}$ defined above, we have $|\mathbf{Pr}[f|_A = \phi] - 2^{-k}| \leqslant 2^{-(m-n)}$, where $f|_A$ denotes the restriction of f to the set A.*

Proof: The k strings in A determine k indices $i_1, \ldots, i_k$ in the truth table of f. The function ϕ is determined by its truth table, a binary string α of length k. Now the claim follows immediately from Lemma 1 and the definition of (ϵ, k)-independence. ■

Razborov [Raz88] showed that there exist complex combinatorial structures (such as the Ramsey graphs, rigid graphs, etc.) of exponential size which can be encoded by polynomial-size bounded-depth Boolean formulas over the basis $\{\&, \oplus, 1\}$. In effect, Razborov gave a construction of ϵ-biased sample spaces (using the terminology of [NN93]), where the elements of such sample spaces are the truth tables of $\mathrm{AC}^0[2]$-computable Boolean functions chosen according to a certain distribution on $\mathrm{AC}^0[2]$-formulas. We describe this distribution next.

For $n, m, l \in \mathbb{N}$, a random formula $\mathbf{F}(n, m, l)$ of depth 3 is defined as

$$\mathbf{F}(n, m, l) = \oplus_{\alpha=1}^{l} \&_{\beta=1}^{m} ((\oplus_{\gamma=1}^{n} \lambda_{\alpha\beta\gamma} x_\gamma) \oplus \lambda_{\alpha\beta}), \tag{1}$$

where $\{\lambda_{\alpha\beta}, \lambda_{\alpha\beta\gamma}\}$ is a collection of $(n+1)ml$ independent random variables uniformly distributed on $\{0,1\}$. The following lemma shows that this distribution determines an ϵ-biased sample space; as observed in [Sav95], a slight modification of the above construction yields somewhat better parameters, but the simpler construction would suffice for us here.

Lemma 3 ([Raz88]) *Let $k, l, m \in \mathbb{N}$ be any numbers such that $k \leqslant 2^{m-1}$, let A be any set of k strings from $\{0,1\}^n$, and let ϕ be any Boolean function defined on A. For a Boolean function f computed by the random formula $\mathbf{F}(n, m, l)$ defined in (1), we have $|\mathbf{Pr}[f|_A = \phi] - 2^{-k}| \leqslant e^{-l2^{-m}}$, where $f|_A$ denotes the restriction of f to the set A.*

The proof of Lemma 3 is most easily obtained by manipulating certain discrete Fourier transforms. We refer the interested reader to [Raz88] or [Sav95] for details.

Below we give the definitions of some classes of Boolean functions hard for 1-b.p.'s. We say that a Boolean function $f_n(x_1, \ldots, x_n)$ is *r-mixed* for some $r \leqslant n$ if, for every subset X of r input variables $\{x_{i_1}, \ldots, x_{i_r}\}$, no two distinct assignments to X yield the same subfunction of f in the remaining $n - r$ variables. We shall see in the following section that an r-mixed function for $r = n - \lceil \log n \rceil - 2$ has a nonzero probability in a distribution $\mathbf{F}_{n,m}$, where $m \in O(n)$, and in the distribution induced by the random formula $\mathbf{F}(n, m, l)$, where $m \in O(\log n)$ and $l \in \mathrm{poly}(n)$.

It was observed by many researchers that r-mixed Boolean functions are hard for 1-b.p.'s. The following lemma is implicit in [Weg88, Dun85], and is a particular case of results in [Juk88, SS93].

Lemma 4 ([Weg88, Dun85, Juk88, SS93]) *Let $f_n(x_1, \ldots, x_n)$ be an r-mixed Boolean function, for some $r \leqslant n$. Then every 1-b.p. computing f_n has size at least $2^r - 1$.*

Following Savický and Žák [SZ96], we call a function $\phi : \{0,1\}^n \to \{1,2,\ldots,n\}$ (s,n,q)-*complete*, for some integers s, n, and q, if for every set $I \subseteq \{1,\ldots,n\}$ of size $n-s$ we have

1. for every 0-1 assignment to the variables x_i, $i \in I$, the range of the resulting subfunction of ϕ is equal to $\{1,2,\ldots,n\}$, and
2. there are at most q different subfunctions of ϕ, as one varies over all 0-1 assignments to x_i, $i \in I$.

Our interest in (s,n,q)-complete functions is justified by the following lemma; its proof is based on a generalization of Lemma 4.

Lemma 5 ([SZ96]) *Let $\phi : \{0,1\}^n \to \{1,2,\ldots,n\}$ be an (s,n,q)-complete function. Then the Boolean function $f_n(x_1,\ldots,x_n) = x_{\phi(x_1,\ldots,x_n)}$ requires 1-b.p.'s of size at least $2^{n-s}/q$.*

The following lemma can be used to construct an (s,n,q)-complete function.

Lemma 6 ([SZ96]) *Let A be a $t \times n$ matrix over* GF(2) *with every $t \times s$ submatrix of rank at least r. Let $\psi : \{0,1\}^t \to \{1,2,\ldots,n\}$ be a mapping such that its restriction to every affine subset of $\{0,1\}^t$ of dimension at least r has the range $\{1,2,\ldots,n\}$. Then the function $\phi(\vec{x}) = \psi(A\vec{x})$ is $(s,n,2^t)$-complete.*

A probabilistic argument shows that a $t \times n$ matrix A and a function $\psi : \{0,1\}^t \to \{1,2,\ldots,n\}$ exist that satisfy the assumptions of Lemma 6 for the choice of parameters $s,t,r \in O(\log n)$, thereby yielding a Boolean function that requires 1-b.p.'s of size at least $2^{n-O(\log n)}$. Below we will show that the argument uses only limited independence of random bits, and hence it can be derandomized using the known constructions of (ϵ,k)-independent spaces. Our proof will utilize the following lemma of Razborov.

Lemma 7 ([Raz88]) *Let $l > 2k$ be any natural numbers, let $0 < \theta, \epsilon < 1$, and let $\mathcal{E}_1,\ldots,\mathcal{E}_l$ be events such that, for every subset $I \subseteq \{1,\ldots,l\}$ of size at most k, we have $|\mathbf{Pr}[\wedge_{i\in I}\mathcal{E}_i] - \theta^{|I|}| \leqslant \epsilon$. Then $\mathbf{Pr}[\wedge_{i=1}^{l}\bar{\mathcal{E}}_i] \leqslant e^{-\theta l} + \binom{l}{k+1}(\epsilon k + \theta^k)$.*

We give the proof of Lemma 7 in Appendix A, since it does not appear to have been translated into English before.

3 Constructing r-Mixed Boolean Functions

First, we give a simple probabilistic argument showing that r-mixed functions exist for $r = n - \lceil \log n \rceil - 2$. Let f be a Boolean function on n variables that is chosen uniformly at random from the set of all Boolean n-variable functions. For any fixed set of indices $\{i_1,\ldots,i_r\} \subseteq \{1,\ldots,n\}$ and any two fixed binary strings $\alpha = \alpha_1,\ldots,\alpha_r$ and $\beta = \beta_1,\ldots,\beta_r$, the probability that fixing $x_{i_1},\ldots,x_{i_r}$ to α and then to β will give the same subfunction of f in the remaining $n-r$ variables is 2^{-k}, where $k = 2^{n-r}$. Thus, the probability that f is not r-mixed is at most $\binom{n}{r}2^{2r}2^{-k}$, which tends to 0 as n grows.

We observe that the above argument only used the fact that f is random on any set of $2k$ inputs: those obtained after the r variables $x_{i_1},\ldots,x_{i_r}$ are fixed to α, the set of which will be denoted as

A_α, plus those obtained after the same variables are fixed to β, the set of which will be denoted as A_β. This leads us to the following theorem.

Theorem 8 *There is an $m \in O(n)$ for which the probability that a Boolean n-variable function f chosen according to the distribution $\mathbf{F}_{n,m}$ is r-mixed, for $r = n - \lceil \log n \rceil - 2$, tends to 1 as n grows.*

Proof: By Lemma 2, the distribution $\mathbf{F}_{n,m}$ yields a function f which is equal to any fixed Boolean function ϕ defined on a set $A_\alpha \cup B_\beta$ of $2k$ inputs with probability at most $2^{-2k} + 2^{-(m-n)}$. The number of functions ϕ that assume the same values on the corresponding pairs of elements $a \in A_\alpha$ and $b \in A_\beta$ is 2^k. Thus, the probability that f is not r-mixed is at most $\binom{n}{r} 2^{2r}(2^{-k} + 2^{-(m-n-k)})$. If $m = (7+\delta)n$ for any $\delta > 0$, then this probability tends to 0 as n grows. ■

By definition, each function from $\mathbf{F}_{n,m}$ can be computed by a Boolean circuit of size $\mathrm{poly}(n, m)$. It must be also clear that checking whether a function from $\mathbf{F}_{n,m}$, given by a $2m$-bit string, is r-mixed can be done in LINSPACE. It follows from Theorem 8 that we can find an r-mixed function, for $r = n - \lceil \log n \rceil - 2$, in LINSPACE by picking the lexicographically first string of $2m$ bits that determines such a function. By Lemma 4, this function will have the optimal lower bound for 1-b.p.'s, $\Omega(2^n/n)$.

We should point out that any of the three constructions of small (ϵ, k)-independent spaces in [AGHP92] could be used in the same manner as described above to obtain an r-mixed Boolean function computable in LINSPACE $\cap$ P/poly, for $r = n - \lceil \log n \rceil - 2$. Applying Lemma 3, we can obtain an r-mixed function with the same value of r.

Theorem 9 *There are $m \in O(\log n)$ and $l \in \mathrm{poly}(n)$ for which the probability that a Boolean n-variable function f computed by the random formula $\mathbf{F}(n, m, l)$ defined in (1) is r-mixed, for $r = n - \lceil \log n \rceil - 2$, tends to 1 as n grows.*

Proof: Proceeding as in the proof of Theorem 8, with Lemma 3 applied instead of Lemma 2, we obtain that the probability that f is not r-mixed is at most $\binom{n}{r} 2^{2r}(2^{-k} + 2^{-(l2^{-m}-k)})$. If $m = \lceil \log n \rceil + 3$ and $l = (6+\delta)n^2$ for any $\delta > 0$, then this probability tends to 0 as n grows. ■

Corollary 10 *There exists a Boolean function computable by a polynomial-size depth 3 formula over the basis $\{\&, \oplus, 1\}$ that requires a 1-b.p. of size at least $\Omega(2^n/n)$ for all sufficiently large n.*

4 Constructing (s, n, q)-Complete Functions

Let us take a look at the probabilistic proof (as presented in [SZ96]) of the existence of a matrix A and a function ψ with the properties assumed in Lemma 6. Suppose that a $t \times n$ matrix A over GF(2) and a function $\psi : \{0,1\}^t \to \{1, 2, \ldots, n\}$ are chosen uniformly at random. For a fixed $t \times s$ submatrix B of A, if $\mathrm{rank}(B) < r$, then there is a set of at most $r-1$ columns in B whose linear span contains each of the remaining $s - r + 1$ columns of B. For a fixed set R of such $r-1$ columns in B, the probability that each of the $s - r + 1$ vectors chosen uniformly at random will be in the linear span of R is at most $(2^{r-1}/2^t)^{s-r+1}$. Thus, the probability that the matrix A is "bad" is at most

$$\binom{n}{s}\binom{s}{r-1} 2^{-(t-r+1)(s-r+1)}. \tag{2}$$

For a fixed affine subspace H of $\{0,1\}^t$ of dimension r and a fixed $1 \leqslant i \leqslant n$, the probability that the range of ψ restricted to H does not contain i is at most $(1 - 1/n)^{2^r}$. The number of

different affine subspaces of $\{0,1\}^t$ of dimension r is at most $2^{(r+1)t}$; the number of different i's is n. Hence the probability that ψ is "bad" is at most

$$2^{(r+1)t} n \left(1 - \frac{1}{n}\right)^{2^r} \leqslant 2^{(r+1)t} n e^{-2^r/n}. \tag{3}$$

An easy calculation shows that setting $s = \lceil (2+\delta)\log n \rceil$, $t = \lceil (3+\delta)\log n \rceil$, and $r = \lceil \log n + 2\log\log n + b \rceil$, for any $\delta > 0$ and sufficiently large b (say, $b = 3$ and $\delta = 0.01$), makes expressions (2) and (3) tend to 0 as n grows.

Theorem 11 *There exist constants $d_1, d_2, d_3 \in \mathbb{N}$ such that every $(2^{-d_1 \log^3 n}, d_2 \log^2 n)$-independent sample space over n^{d_3}-bit strings contains both matrix A and function ψ with the properties as in Lemma 6, for $s, r, t \in O(\log n)$.*

Proof: We observe that both probabilistic arguments used only partial independence of random bits. For A, we need a tn-bit string coming from an (ϵ, k)-independent sample space with $k = ts$ and $\epsilon = 2^{-c_1 \log^2 n}$, for a sufficiently large constant c_1. Indeed, for a fixed $t \times s$ submatrix B of A and a fixed set R of $r-1$ columns in B, the number of "bad" $t \times s$-bit strings α filling B so that the column vectors in R contain in their linear span all the remaining $s-r+1$ column vectors of B is at most $2^{(r-1)t} 2^{(r-1)(s-r+1)} = 2^{(r-1)(s+t-r+1)}$. If A is chosen from the (ϵ, k)-independent sample space with ϵ and k as above, then the probability that some fixed "bad" string α is chosen is at most $2^{-ts} + \epsilon$. Thus, in this case, the probability that A is "bad" is at most

$$\binom{n}{s}\binom{s}{r-1}(2^{-(t-r+1)(s-r+1)} + \epsilon 2^{(r-1)(s+t-r+1)}).$$

Choosing the same s, t, and r as in the case of fully independent probability distribution, one can make this probability tend to 0 as n grows, by choosing sufficiently large c_1.

Similarly, for the function ψ, we need a $2^t\lceil \log n \rceil$-bit string from an (ϵ, k)-independent sample space with $k = c_2 \log^2 n$ and $\epsilon = 2^{-c_3 \log^3 n}$, for sufficiently large constants c_2 and c_3. Here we view the truth table of ψ as a concatenation of 2^t $\lceil \log n \rceil$-bit strings, where each $\lceil \log n \rceil$-bit string encodes a number from $\{1, \ldots, n\}$. The proof, however, is slightly more involved in this case, and depends on Lemma 7.

Let s, r, and t be the same as before. For a fixed affine subspace $H \subseteq \{0,1\}^t$ of dimension r, such that $H = \{a_1, \ldots, a_l\}$ for $l = 2^r$, and for a fixed $1 \leqslant i \leqslant n$, let $\mathcal{E}_j$, $1 \leqslant j \leqslant l$, be the event that $\psi(a_j) = i$ when ψ is chosen from the (ϵ, k)-independent sample space defined above. Then Lemma 7 applies with $\theta = 2^{-\lceil \log n \rceil}$, yielding that the probability that ψ misses the value i on the subspace H is

$$\mathbf{Pr}[\wedge_{j=1}^{l} \bar{\mathcal{E}}_j] \leqslant e^{-2^{r-\lceil \log n \rceil}} + \binom{2^r}{k+1}(\epsilon k + 2^{-k\lceil \log n \rceil}). \tag{4}$$

It is easy to see that the first term on the right-hand side of (4) is at most $e^{-4\log^2 n}$ (when $b = 3$ in r). We need to bound from above the remaining two terms: $\binom{2^r}{k+1} 2^{-k\lceil \log n \rceil}$ and $\binom{2^r}{k+1}\epsilon k$. Using Stirling's formula, one can show that the first of these two terms can be made at most $2^{-4\log^2 n}$, by choosing c_2 sufficiently large. Having fixed c_2, we can also make the second of the terms at most $2^{-4\log^2 n}$, by choosing $c_3 > c_2$ sufficiently large. It is then straightforward to verify that the probability that ψ misses at least one value i, $1 \leqslant i \leqslant n$, on at least one affine subspace of dimension r tends to 0 as n grows. ■

Using any efficient construction of almost independent sample spaces, for example, Pow_N^{2m} with $N = tn \in O(n \log n)$ and $m \in O(\log^2 n)$, we can find a matrix A with the required properties in $\mathrm{DTIME}(2^{O(\log^2 n)})$ by searching through all elements of the sample space and checking whether any of them yields a desired matrix. Analogously, we can find the required function ψ in $\mathrm{DTIME}(2^{O(\log^3 n)})$, by considering, e.g., $\mathrm{Pow}_{N'}^{2m'}$ with $N' = 2^t\lceil \log n \rceil$ and $m' \in O(\log^3 n)$. Thus, constructing both A and ψ can be carried out in quasipolynomial time.

Given the corresponding advice strings of $O(\log^3 n)$ bits, ψ is computable in time $\mathrm{polylog}(n)$ and all elements of A can be computed in time $n\mathrm{polylog}(n)$. So, in this case, the function $\phi(\vec{x}) = \psi(A\vec{x})$ is computable in quasilinear time. Hence, by "hard-wiring" good advice strings, we get the function $f_n(\vec{x}) = x_{\phi(\vec{x})}$ computable by quasilinear-size circuits, while, by Lemmas 5 and 6, f_n requires 1-b.p.'s of size at least $2^{n-(5+\epsilon)\log n}$, for any $\epsilon > 0$ and sufficiently large n; these parameters appear to be better than those in [ABCR97]. By making the advice strings a part of the input, we obtain a function in QL that requires 1-b.p.'s of size at least $2^{n-O(\log^3 n)}$.

We end this section by observing that the method used above to construct an (s, n, q)-complete Boolean function could be also used to construct an r-mixed Boolean function for $r = n - O(\log n)$ by derandomizing Savický's [Sav99] modification of the procedure in [SZ96]. This r-mixed function is also determined by an advice string of length $\mathrm{polylog}(n)$, and hence can be constructed in quasipolynomial time.

5 Concluding Remarks

We have shown how the known constructions of small ϵ-biased sample spaces [Raz88, NN93, AGHP92] can be directly used to obtain Boolean functions that are exponentially hard for 1-b.p.'s. One might argue, however, that the hard Boolean functions constructed in Sections 3 and 4 are not "explicit" enough, since they are defined as the lexicographically first functions in certain search spaces. It would be interesting to find a Boolean function in P or NP with the optimal lower bound $\Omega(2^n/n)$ for 1-b.p.'s. The problem of constructing a polynomial-time computable r-mixed Boolean function with r as large as possible is of independent interest; at present, the best such function is given in [SZ96] for $r = n - \Omega(\sqrt{n})$. A related open question is to determine whether the minimum number of bits needed to specify a Boolean function with the optimal lower bound for 1-b.p.'s, or an r-mixed Boolean function for $r = n - \lceil \log n \rceil - 2$, can be sublinear.

Acknowledgements. I am indebted to Alexander Razborov for bringing [Raz88] to my attention. I would like to thank Stephen Cook and Petr Savický for their comments on a preliminary version of this paper, and Dieter van Melkebeek for helpful discussions. Finally, I want to express my sincere gratitude to Stephen Cook for his constant encouragement and support.

References

[ABCR97] A.E. Andreev, J.L. Baskakov, A.E.F. Clementi, and J.D.P. Rolim. Small pseudo-random sets yield hard functions: New tight explicit lower bounds for branching programs. *Electronic Colloquium on Computational Complexity*, TR97-053, 1997.

[ABH+86] M. Ajtai, L. Babai, P. Hajnal, J. Komlós, P. Pudlak, V. Rödl, E. Szemerédi, and G. Turán. Two lower bounds for branching programs. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 30–38, 1986.

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost $k-$wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992. (preliminary version in FOCS'90).

[BW98] B. Bollig and I. Wegener. A very simple function that requires exponential size read-once branching programs. *Information Processing Letters*, 66:53–57, 1998.

[Dun85] P.E. Dunne. Lower bounds on the complexity of one-time-only branching programs. In L. Budach, editor, *Proceedings of the Second International Conference on Fundamentals of Computation Theory*, volume 199 of *Lecture Notes in Computer Science*, pages 90–99, Springer Verlag, Berlin, 1985.

[Gal97] A. Gal. A simple function that requires exponential size read-once branching programs. *Information Processing Letters*, 62:13–16, 1997.

[Juk88] S. Jukna. Entropy of contact circuits and lower bound on their complexity. *Theoretical Computer Science*, 57:113–129, 1988.

[KMW91] M. Krause, C. Meinel, and S. Waack. Separating the eraser Turing machine classes L_e, NL_e, $\mathrm{co-NL}_e$ and P_e. *Theoretical Computer Science*, 86:267–275, 1991.

[NN93] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993. (preliminary version in STOC'90).

[Pon99] S. Ponzio. A lower bound for integer multiplication with read-once branching programs. *SIAM Journal on Computing*, 28(3):798–815, 1999. (preliminary version in STOC'95).

[Raz88] A.A. Razborov. Bounded-depth formulae over $\{\&, \oplus\}$ and some combinatorial problems. In S. I. Adyan, editor, *Problems of Cybernetics. Complexity Theory and Applied Mathematical Logic*, pages 149–166. VINITI, Moscow, 1988. (in Russian).

[Raz91] A.A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In L. Budach, editor, *Proceedings of the Eighth International Conference on Fundamentals of Computation Theory*, volume 529 of *Lecture Notes in Computer Science*, pages 47–60, Springer Verlag, Berlin, 1991.

[Sav95] P. Savický. Improved Boolean formulas for the Ramsey graphs. *Random Structures and Algorithms*, 6(4):407–415, 1995.

[Sav99] P. Savický. personal communication, January 1999.

[SS93] J. Simon and M. Szegedy. A new lower bound theorem for read-only-once branching programs and its applications. In J.-Y. Cai, editor, *Advances in Computational Complexity*, pages 183–193. AMS-DIMACS Series, 1993.

[SZ96] P. Savický and S. Zák. A large lower bound for 1-branching programs. *Electronic Colloquium on Computational Complexity*, TR96-036, 1996.

[Weg88] I. Wegener. On the complexity of branching programs and decision trees for clique function. *Journal of the ACM*, 35:461–471, 1988.

[Zak84] S. Zak. An exponential lower bound for one-time-only branching programs. In *Proceedings of the Eleventh International Symposium on Mathematical Foundations of Computer Science*, volume 176 of *Lecture Notes in Computer Science*, pages 562–566, Springer Verlag, Berlin, 1984.

A Proof of Lemma 7

We first consider the case where k is even. Let $\mathcal{C}_1, \ldots, \mathcal{C}_l$ be independent events, each having the success probability θ. Applying the Boole-Bonferroni inequality to $\mathbf{Pr}[\vee_{i=1}^{l} \mathcal{E}_i]$ and $\mathbf{Pr}[\vee_{i=1}^{l} \mathcal{C}_i]$, we obtain that

$$\mathbf{Pr}[\vee_{i=1}^{l} \mathcal{E}_i] \geqslant \sum_{\nu=1}^{k} (-1)^{\nu+1} \sum_{|I|=\nu} \mathbf{Pr}[\wedge_{i \in I} \mathcal{E}_i] \tag{5}$$

and

$$\mathbf{Pr}[\vee_{i=1}^{l} \mathcal{C}_i] \leqslant \sum_{\nu=1}^{k} (-1)^{\nu+1} \sum_{|I|=\nu} \theta^{|I|} + \sum_{|I|=k+1} \theta^{k+1}. \tag{6}$$

The assumption of the lemma that $\mathcal{E}_1, \ldots, \mathcal{E}_l$ are almost k-wise independent implies that the right-hand side in (5) is at least

$$\sum_{\nu=1}^{k} (-1)^{\nu+1} \sum_{|I|=\nu} \theta^{|I|} - \epsilon k \binom{l}{k}. \tag{7}$$

On the other hand, the independence of $\mathcal{C}_1, \ldots, \mathcal{C}_l$ implies that

$$\mathbf{Pr}[\vee_{i=1}^{l} \mathcal{C}_i] = 1 - (1-\theta)^l \geqslant 1 - e^{-\theta l}. \tag{8}$$

Combining (5), (7), (6), and (8) yields (for even k) that

$$\begin{aligned} \mathbf{Pr}[\vee_{i=1}^{l} \mathcal{E}_i] &\geqslant 1 - e^{-\theta l} - \epsilon k \binom{l}{k} - \theta^{k+1} \binom{l}{k+1} \\ &\geqslant 1 - e^{-\theta l} - \binom{l}{k+1} (\epsilon k + \theta^{k+1}). \end{aligned}$$

In the case where k is odd, we use the above argument with $k-1$ substituted for k. This completes the proof.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'