Electronic Colloquium on Computational Complexity, Report No. 155 (2005)

Constructions of low-degree and error-correcting ϵ-biased sets

Amir Shpilka*

Abstract

In this work we give two new constructions of ϵ-biased generators. Our first construction answers an open question of Dodis and Smith [DS05], and our second construction significantly extends a result of Mossel et al. [MST03]. In particular we obtain the following results:

1. We construct a family of asymptotically good binary codes such that the codes in our family are also ϵ-biased sets for an exponentially small ϵ. Our encoding and decoding algorithms run in polynomial time in the block length of the code. This answers an open question of Dodis and Smith [DS05].
2. We construct a degree k ϵ-biased generator, $G : \{0,1\}^m \to \{0,1\}^n$, for every $k = o(\log n)$. For k constant we get that $n = \Omega(m/log(1/\epsilon))^k$, which is nearly optimal. Our result also separates degree k generators from generators in NC^0_k, showing that the stretch of the former can be much larger than the stretch of the latter. This problem of constructing degree k generators was introduced by Mossel et al. [MST03] who gave a construction only for the case of degree 2 generators.

1 Introduction

A subset $S \subset \{0,1\}^n$ is called an ϵ-biased set if its bias with respect to any linear test is at most ϵ. Namely, for every non-zero vector $w \in \{0,1\}^n$ we have that $|\Pr_{s\in S}[\langle w, s\rangle = 1] - 1/2| \leq \epsilon$. In other words, for every hyperplane $H \subset \{0,1\}^n$ it holds that $\left||S \cap H| - \frac{|S|}{2}\right| \leq \epsilon|S|$. An ϵ-biased generator is a mapping $G : \{0,1\}^m \to \{0,1\}^n$ whose image is an ϵ-biased set. A subset $C \subset \{0,1\}^n$ is called a good error correcting code[1] if it has an exponential size and the Hamming distance between any two of its elements is linear.

In this paper we give two constructions of ϵ-biased sets. The first is a construction of an ϵ-biased generator such that each of its output bits is a low degree polynomial. Our second

*Faculty of Computer Science, The Technion, Haifa 32000, Israel. Email: shpilks@cs.technion.ac.il.

[1]To be completely accurate we have to speak about family of codes, and we do it in a later section.

ISSN 1433-8092

result is a construction of a family of (efficiently encodable and decodable) error correcting codes that are also ϵ-biased sets for an exponentially small ϵ.

Background. The notion of ϵ-biased sets (or more accurately of an ϵ-biased distribution) was first defined by Naor and Naor [NN93] who also gave the first constructions of such distributions and demonstrated the power of ϵ-biased sets for several applications. The construction of Naor and Naor was later improved in a series of papers [AGHP92, AIK$^+$90, RSW93, EGL$^+$92, AM95]. Since their first appearance ϵ-biased sets have found many applications in different areas of theoretical computer science including: derandomization of algorithms such as fast verification of matrix multiplication [NN93]; construction of almost k-wise independent distributions [NN93, MNN94]; inapproximability results for quadratic equations over GF(2) [HPS93]; learning theory [AM95]; explicit constructions of Ramsey graphs [Nao92]; explicit constructions of Cayley expanders [AR94, MW02]; construction of efficient low degree tests and short PCPs [BSSVW03]; and construction of two-source extractors [Raz05].

In several recent works ϵ-biased sets were studied from a different perspective. In [CM01] Cryan and Miltersen ask whether there exist an NC^0 construction of an ϵ-biased generator for which n is super-linear in m. This question was answered affirmatively by Mossel et al. [MST03] who gave a construction of a generator in NC^0_k with $n = m^{\Omega(\sqrt{k})}$. Mossel et al. also raised the question of constructing degree k ϵ-biased generators - i.e. generators that each of their output bits is a degree k polynomial in the input bits. They were also able to give a construction of a degree 2 generator with a near optimal stretch (i.e. $n = \Omega(m^2)$ and $\epsilon = \exp(-O(n))$).

In [DS05] Dodis and Smith ask "Does there exist an explicitly-constructible ensemble of good codes with small bias and polytime encoding and decoding algorithms (ideally, codes with linear rate and minimum distance, and negligible bias)?". Namely, Dodis and Smith raise the question of constructing a family of good codes that are also ϵ-biased sets for an exponentially (in the block length of the code) small ϵ. Such a family of codes was needed for the construction of a cryptographic scheme that will enable two parties to securely correct errors in a shared secret string. Dodis and Smith managed to give a protocol for correcting errors without leaking information without constructing a family of ϵ-biased good codes, however such a construction can simplify their scheme.

Our results. Our first result is a construction of degree k ϵ-biased generators of maximal stretch (up to a constant). Namely we give a construction of a degree k generator from m bits to $n = \Omega((m/\log(1/\epsilon))^k)$ bits, for any fixed k. Thus, for every fixed ϵ the output length is $\Omega(m^k)$, and clearly the output length cannot exceed m^k (as there are only $O(m^k)$ linearly independent polynomials of degree k in m variables).

Theorem 1 *For every integer $0 < k$ and every large enough[2] integer m and every $\epsilon >$*

[2]More precisely, there exists m_0, independent of k, such that for every $m \geq m_0$...

$\exp(-O(\frac{m^{1-\frac{1}{k}}}{k2^k}))$, *there is a mapping* $G : \{0,1\}^m \to \{0,1\}^n$, *where* $n = \Omega((\frac{m}{k2^k \log(1/\epsilon)})^k)$, *such that* G *is a degree* k *generator with bias at most* ϵ.

We note that as k is constant the output is of length $\Omega((m/\log(1/\epsilon))^k)$. In particular this gives a separation between ϵ-biased generators in NC^0_k and degree k generators: Theorem 6 of [MST03] shows that the stretch (i.e. $n-m$) of any ϵ-biased generator in NC^0_k, for $\epsilon < 2^{-2k}$, is at most $O\left(2^k m^{\lceil k/2 \rceil}\right)$. In contrast we can get a stretch of $\Omega((\frac{m}{k2^k \log(1/\epsilon)})^k)$ that for a fixed k and a not too small ϵ (say $\epsilon = 2^{-m^{o(1)}}$) is at least $m^{k-o(1)}$.

Our second result is a construction of a family of ϵ-biassed good codes. Namely, we give a construction of a family of good codes (constant relative rate, constant relative distance, efficient encoding and decoding algorithms) such that the codes in the family have an exponential small bias. Thus our construction answers affirmatively the open question of [DS05].

Theorem 2 *Let* $a = 0.0595/2$. *Then for every large enough integers* $0 \leq t, n$ *where* $n = 189 \cdot 8^t$ *there is a polynomial time constructible generator* $G : \{0,1\}^m \to \{0,1\}^n$, *where* $m \geq an/48$, *such that its image* $C = G(\{0,1\}^m)$ *is a code with the following properties.* C *has relative rate* $\geq a/48$; *relative distance* $\geq a^2/24$; *a polynomial time decoding algorithm that can fix* $a^2/48$ *fraction of errors; and the bias of* C *(and hence of* G*) is* $\epsilon = \exp(-O(n))$.

Motivation. One prosaic motivation for studying these questions is that they were studied before and remained open. However, we feel that there is a stronger argument in favor of studying these problems: both problems are very natural and combine notions that were found to have many applications in theoretical computer science.

It is very desirable to give explicit constructions of low complexity for combinatorial objects. For example, in [Hås87, Gol00a, CM01, KL01, MST03, AIK04] questions regarding the existence of objects such as ϵ-biased generators, one-way functions and pseudo-random generators in NC^0 were studied. As low degree polynomial are a natural "low complexity" class we feel that constructing low degree ϵ-biased generators is a natural question.

Error correcting codes (ECCs) have many applications in theoretical computer science (cf. [Fei95, Gur01, Tre04]). Finding explicit constructions of ECCs is an extensively studied question (cf. [MS77, Tre04]). In recent years the focus in the theoretical computer science community is on giving explicit constructions of ECCs that have additional properties, for example: codes that have efficient list-decoding algorithms (cf. survey of Trevisan [Tre04]), quantum codes (cf. [NC00, KSV02]), codes that are locally testable and codes that are locally decodable (cf. survey of Goldreich [Gol00b]). We believe that as error correcting codes and ϵ-biased sets are such important objects it is natural to combine them and to construct an ϵ-biased error correcting codes. We expect that our constructions will find new applications.

Methods. Our constructions are similar in spirit to the constructions of ϵ-biased generators of Mossel et al. [MST03]. In the proofs of both Theorem 1 and Theorem 2 we first construct

a generator $G^{(h)}$ that is (almost) unbiased w.r.t. heavy tests[3], i.e. the bias of $G^{(h)}$ w.r.t. any test w of large weight is small. Then we construct a generator $G^{(l)}$ that is (almost) unbiased w.r.t. light tests and then we take their XOR, on two independent inputs, to get the final generator $G(x,y) = G^{(h)}(x) \oplus G^{(l)}(y)$. A significant difference from [MST03] is in the way that we construct $G^{(l)}$. Previously $G^{(l)}$ was constructed "from scratch", i.e. there was no connection between the construction of $G^{(l)}$ and of $G^{(h)}$. We show a novel use of error correcting codes that enables us to transform *any* generator that is (almost) unbiased with respect to heavy tests to a generator that is (almost) unbiased w.r.t. light tests. In the heart of this transformation lies the observation that from the generating matrix of a linear error correcting code of relative rate 1/2 one can construct a linear transformation, from $\{0,1\}^n$ to itself, that sends "light" vectors to "heavy" vectors (where "light" and "heavy" depend on the properties of the code).

Organization. In Section 2 we give the basic notations and definitions. In particular in subsection 2.1 we give the basic definitions of error correcting codes and recall a construction of self-dual error correcting codes, and in subsection 2.2 we give the basic definitions regarding ϵ-biased sets and ϵ-biased generators and give the proofs of some well known facts. In section 3 we prove Theorem 1, and in section 4 we prove Theorem 2.

2 Preliminaries

We shall denote with $\log(x)$ the natural logarithm of x, i.e. $\log(x) = \log_e(x)$. We denote $\exp(x) = e^x$. We use the notation of $|I|$ to denote the size of the set I. For a random variable X that is distributed according to a distribution D we denote with $\mathbb{E}_D[X]$ the expectation of X. For a matrix A we denote with A^t the transpose of A.

For a vector $v \in \{0,1\}^n$ we denote with v_i the i'th coordinate of v. Namely, $v = (v_1, \ldots, v_n)$. For two vectors $u, v \in \{0,1\}^n$ we denote with $\mathrm{dist}(u,v)$ the hamming distance between u and v, i.e. $\mathrm{dist}(u,v) = |\{i : u_i \neq v_i\}|$. We also denote with $\mathrm{wt}(v)$ (the weight of v) the number of non-zero coordinates of v. In other words, $\mathrm{wt}(v) = \mathrm{dist}(v, \vec{0})$, where $\vec{0}$ is the zero vector. For $v, u \in \{0,1\}^n$ we denote with $\langle v, u \rangle$ their inner product modulo 2, i.e. $\langle v, u \rangle = v_1 \cdot u_1 \oplus \ldots \oplus v_n \cdot u_n$.

For two vectors $v, u \in \{0,1\}^n$ we denote $v \oplus u = (v_1 \oplus u_1, \ldots, v_n \oplus u_n)$, i.e. it is the coordinate-wise XOR of v and u. For two multisets $S_1, S_2 \subseteq \{0,1\}^n$ we denote $S_1 \oplus S_2 = \{v \oplus u | v \in S_1 \text{ and } u \in S_2\}$. For two functions $G_1 : \{0,1\}^{m_1} \to \{0,1\}^n$ and $G_2 : \{0,1\}^{m_2} \to \{0,1\}^n$ we denote with $G = G_1 \oplus G_2$ the function $G(x,y) : \{0,1\}^{m_1} \times \{0,1\}^{m_2} \to \{0,1\}^n$ satisfying $G(x,y) = G_1(x) \oplus G_2(y)$. Usually we write $\{0,1\}^{m_1+m_2}$ instead of $\{0,1\}^{m_1} \times \{0,1\}^{m_2}$.

[3] As mentioned earlier, every test can be identified with a binary vector of length n. The weight of the test is the number of non zero coordinates of this vector.

2.1 Error correcting codes

Let $E : \{0,1\}^k \to \{0,1\}^n$. Denote with $C = E(\{0,1\}^n)$ the image of E. Then C is called an $[n,k,d]$-code if for any two codewords $E(v), E(u) \in C$, where $u \neq v$, we have that $\mathrm{dist}(E(u), E(v)) \geq d$. We denote with $R = k/n$ the relative rate of C and with $\delta = d/n$ the relative minimal distance of C, and say that C is an $[R,\delta]$-code. When E is a linear mapping we say that C is a linear code. A map $D : \{0,1\}^n \to \{0,1\}^k$ can correct t errors if for any $v \in \{0,1\}^k$ and any $w \in \{0,1\}^n$ such that $\mathrm{dist}(E(v), w) \leq t$ we have that $D(w) = v$. Such a D is called a decoding algorithm for C. A family of codes $\{C_i\}$, where C_i is an $[R_i, \delta_i]$-code of block length n_i, has constant rate if there exists a constant $0 < R$ such that for all codes in the family it holds that $R_i \geq R$. The family has a linear distance if there exists a constant $0 < \delta$ such that for all codes in the family we have $\delta_i \geq \delta$. In such a case we say that the family is a family of $[R,\delta]$ codes. If a family of codes as above has $\lim_{i\to\infty} n_i = \infty$, a constant rate and a linear minimal distance then we say that the family is a family of good codes and that the codes in the family are good. Similarly, we say that the family of codes has a decoding algorithm for a fraction τ of errors if for each C_i there is a decoding algorithm D_i that can decode from $\tau \cdot n_i$ errors.

When C is a linear code we define the dual of C in the following way:

$$C^{\perp} \stackrel{\Delta}{=} \{y \in \{0,1\}^n : \forall x \in C \ \ \langle y, x\rangle = 0\}.$$

A family of codes is said to have good dual codes if the family of the dual codes is good.

In our constructions we will need to use a family of good error correcting codes with good dual codes. The following result of Matsumoto[Mat02] (after a result of Ashikhmin et al. [ALT01]) gives an explicit construction of such a family. As this result is implicit in [Mat02] we give a sketch of the proof.

Theorem 3 (Matsumoto) *For an integer t let $n_t = \frac{189}{4}8^t$. Then, for a large enough t, there is a polynomial time constructible self-dual $[\frac{1}{2}, 0.0595]$-code of block length n_t. Moreover, the code has a polynomial time (in the block length) decoding algorithm that can correct from $> 0.0595/2$ fraction of errors.*

Proof In the proof we use the notations of [Mat02]. Consider proposition 2 of [Mat02]. Let $m = 3$ then $n_t = \frac{63}{8}8^t$. Now let $j = 0$ and t large enough so that $\frac{g_t}{n_t} \leq \frac{1}{7} + 0.0001$. Then the code $C_t = C(G_0 + 0 \cdot P_\infty)$ is an $[n_t, n_t/2, n_t/2 - n_t/7 - 0.0001n_t]$ *self-dual* code over the field $\mathrm{GF}(2^6)$. By expanding this code w.r.t. a self dual basis of $\mathrm{GF}(2^6)$ over $\mathrm{GF}(2)$ (see section III of [ALT01]), we get a $[6n_t, 3n_t, n_t/2 - n_t/7 - 0.0001n_t]$ binary code D_t. The block length is thus $\frac{189}{4}8^t$, the relative rate of this code is $1/2$ and its relative minimum distance is larger than 0.0595. As C_t was self-dual then so is D_t. The fact regarding the decoding algorithm follows from the work of Feng and Rao [FR93]. □

2.2 ϵ-biased sets

Definition 4 *Let $S \subseteq \{0,1\}^n$ be a set (or a multi-set). For a non zero vector $w \in \{0,1\}^n$ we denote with* $\mathrm{bias}_w(S)$ *the bias of S w.r.t. w. That is,*

$$\mathrm{bias}_w(S) \stackrel{\Delta}{=} \left|\frac{1}{2} - \Pr_{s\in S}[\langle w,s\rangle = 1]\right| = \left|\frac{1}{2} - \frac{1}{|S|}\sum_{s\in S}\langle w,s\rangle\right|.$$

The bias of S is equal to the maximal bias w.r.t. any non-zero test:

$$\mathrm{bias}(S) = \max_{\vec{0}\neq w\in\{0,1\}^n} \mathrm{bias}_w(S).$$

In particular, S is ϵ-biased if for every $\vec{0} \neq w \in \{0,1\}^n$ it holds that $\mathrm{bias}_w(S) \leq \epsilon$.

In order to define ϵ-biased generator it will be convenient to speak about ϵ-biased distributions. In the following we identify vectors $w \in \{0,1\}^n$ with a subset $W \subseteq [n]$ in the usual manner ($i \in W$ if and only if $w_i = 1$).

Definition 5 *Let $G = (G_1, \ldots, G_n)$ be a random variable ranging over $\{0,1\}^n$. The bias of G w.r.t. a non-zero vector $w \in \{0,1\}^n$ is defined to be*

$$\mathrm{bias}_w(G) \stackrel{\Delta}{=} \left|\frac{1}{2} - \Pr_{x\in\{0,1\}^n}[\oplus_{i\in W} G_i(x) = 1]\right| = \left|\frac{1}{2} - \mathbb{E}_{x\in\{0,1\}^n}\langle G(x), w\rangle\right|.$$

The bias of G is equal to the maximal bias w.r.t. any non-zero test:

$$\mathrm{bias}(G) = \max_{\vec{0}\neq w\in\{0,1\}^n} \mathrm{bias}_w(G).$$

In particular, G is ϵ-biased if for every non-zero w it holds that $\mathrm{bias}_w(G) \leq \epsilon$.

When $G : \{0,1\}^m \to \{0,1\}^n$ is a map we define the bias of G to be the bias of the random variable $G = (G_1, \ldots, G_n)$, where G_i is the i'th output bit of G. Notice that the bias of G is equal to the bias of the multi-set $G(\{0,1\}^m)$.

The following well known lemma gives information about the bias of the XOR of two independent random variables (or the XOR of two different sets).

Lemma 6 *Let G_1, G_2 be two independent random variables taking values in $\{0,1\}^n$. Then* $\mathrm{bias}(G_1 \oplus G_2) \leq 2\mathrm{bias}(G_1)\mathrm{bias}(G_2) \leq \min_i \mathrm{bias}(G_i)$. *Similarly, if $S_1, S_2 \subseteq \{0,1\}^n$ we have that* $\mathrm{bias}(S_1 \oplus S_2) \leq 2\mathrm{bias}(S_1)\mathrm{bias}(S_2) \leq \min_i \mathrm{bias}(S_i)$.

Proof We give the proof only for the case of two random variables as the proof for two sets is basically the same. Fix $\vec{0} \neq w \in \{0,1\}^n$. As G_1 and G_2 are independent it follows that

$$\Pr[\oplus_{i\in S_w}(G_1 \oplus G_2)_i = 1] =$$
$$\Pr[(\oplus_{i\in S_w}(G_1)_i) = 1] + \Pr[(\oplus_{i\in S_w}(G_2)_i) = 1] - 2\Pr[(\oplus_{i\in S_w}(G_1)_i) = 1]\cdot\Pr[(\oplus_{i\in S_w}(G_2)_i) = 1].$$

In particular we get that

$$\begin{aligned}\mathrm{bias}_w(G_1 \oplus G_2) &= \left|\frac{1}{2} - \Pr[\oplus_{i\in S_w}(G_1 \oplus G_2)_i = 1]\right| \\ &= \left|2\left(\frac{1}{2} - \Pr[\oplus_{i\in S_w}(G_1)_i = 1]\right)\left(\frac{1}{2} - \Pr[\oplus_{i\in S_w}(G_2)_i = 1]\right)\right| \\ &= 2\mathrm{bias}_w(G_1)\mathrm{bias}_w(G_2).\end{aligned}$$

As it is always the case that $\mathrm{bias}_w(G_i) \leq 1/2$, it follows that for every $\vec{0} \neq w \in \{0,1\}^n$ we have that $\mathrm{bias}_w(G_1 \oplus G_2) \leq \min(\mathrm{bias}_w(G_1), \mathrm{bias}_w(G_2))$, and the claim easily follows. □

The following lemma is an immediate corollary of Lemma 6.

Lemma 7 *Let $(G_i)_{i\in I}$ be independent random variables taking values in $\{0,1\}^n$. Then*

$$\mathrm{bias}(\oplus_{i\in I} G_i) \leq 2^{|I|-1}\prod_{i\in I}\mathrm{bias}(G_i).$$

Similarly, if $(S_i)_{i\in I}$ is a family of subsets of $\{0,1\}^n$ we have that

$$\mathrm{bias}(\oplus_{i\in I} S_i) \leq 2^{|I|-1}\prod_{i\in I}\mathrm{bias}(S_i).$$

As a special case of this lemma we get the well known estimate for the bias of a sum of independent random coins.

Lemma 8 *Let $X_1, \ldots, X_t$ be independent 0/1 random variables. Assume that for some $0 < \delta < 1/2$ and for every i we have that $\delta \leq \Pr[X_i = 1] \leq 1 - \delta$, then*

$$\mathrm{bias}\left(\oplus_{i=1}^t X_i\right) \leq \frac{1}{2}\left(1 - 2\delta\right)^t.$$

Another basic fact that we shall need is an estimate on the bias of a degree k polynomial, which follows immediately from the famous Schwartz-Zippel theorem [Sch80, Zip79].

Lemma 9 (Schwartz-Zippel) *Let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant degree k polynomial, then*

$$\frac{1}{2^k} \leq \Pr[f(x) = 1] \leq 1 - \frac{1}{2^k}.$$

Equivalently,

$$\mathrm{bias}\left(f(x)\right) \leq \frac{1}{2} - \frac{1}{2^k}.$$

3 Low degree ϵ-biased generator

In this section we construct an ϵ-bias generator where each of its output bits is a low degree polynomial. Similarly to the construction in the paper of Mossel et al. [MST03] our generator is the combination of two other generators. One generator has a low bias w.r.t. "heavy" tests (tests that involve a large number of output bits) and the other has a low bias against "light" tests (i.e. tests that involve a small number of output bits). The final generator is obtained by XOR-ing the output of the two generators on independent seeds.

We first construct a generator that is unbiased against heavy tests, and then show a general method for transforming generators that are unbiased with respect to heavy tests into generators that are unbiased with respect to light tests.

3.1 Construction for heavy tests

The following theorem gives a construction of a generator that is unbiased w.r.t. heavy tests. We try to give the most general statement so it is a bit cumbersome. The reader should have the following "relaxed" statement in mind: For every k, m and every $0 < \epsilon$ (such that ϵ is not too small as a function of m, say $\epsilon > 2^{-\sqrt{m}}$), we can construct a degree k ϵ-biased generator (that is unbiased w.r.t. heavy tests) from m bits to n bits where $n = \Omega((m/k\log(1/\epsilon))^k)$. The difference of the relaxed statement from the formal one, is that we need to make all the parameters involved in the construction integers, and we aim to make the bound on ϵ as small as possible. We now give the formal statement of the theorem.

Theorem 10 *For every two integers k, m and $0 < a, \epsilon$ such that $\epsilon \geq \exp(-O(\frac{am^{1-\frac{1}{k}}}{k2^k}))$, we define $\gamma = \gamma_{k,\epsilon,a} = \frac{2a}{2^k\log(1/2\epsilon)+4a}$, $s_0 = \lfloor \gamma m \rfloor$, $i_0 = \lfloor m/s_0 \rfloor$. Assume that n is such that $\frac{1}{8} i_0 \binom{s_0}{k} \leq n \leq i_0 \binom{s_0}{k}$. Then there is an explicit map $G^{(h)} : \{0,1\}^m \to \{0,1\}^n$ with the following properties:*

- *For every $w \in \{0,1\}^n$ such that $\mathrm{wt}(w) \geq an$ we have that $\mathrm{bias}_w\left(G^{(h)}\right) \leq \epsilon$.*
- *Each output bit of $G^{(h)}$ is a degree k polynomial in the input bits.*

Proof We first give a sketch of the proof. Partition the m input bits to k sets of roughly equal sizes. For each subset of inputs B we define a set of output bits that depend only on the input bits in B. Thus, output bits that correspond to different sets are independent (as random variables when the input is chosen uniformly at random from $\{0,1\}^n$). The output bits that depend on a given set of inputs are in $1-1$ correspondence with multilinear monomials of degree k. Namely, every output bit is the evaluation of a degree k monomial on the input bits of the relevant set. Now, given a "heavy" linear combination of the output bits, we can present it as a linear combination of "many" degree k polynomials, where each polynomial is defined on a different subset of inputs. As these polynomials are defined on distinct sets of

variables, they are independent random variables, and by Lemma 8 the bias of their sum is small. We now give the more formal proof.

Denote with $\mathcal{M}_k[y_1, \ldots, y_{s_0}]$ the set of all degree k monomials in s_0 variables. The size of $\mathcal{M}_k$ is $\binom{s_0}{k}$. Partition the input bits $\{x_1, \ldots, x_m\}$ to i_0 sets of size s_0, and to a leftover set of size $m - i_0 \cdot s_0 < s_0$. The i-th set is $B_i = \{x_{s_0 \cdot (i-1)+1}, \ldots, x_{s_0 \cdot i}\}$. We first deal with the case that $n = i_0\binom{s_0}{k}$: For every one of the first i_0 sets of the partition we define $\binom{s_0}{k}$ output bits. Each of the output bits corresponds to a different monomial from $\mathcal{M}_k$ evaluated on the variables of B_i. We denote with G_i the set of output bits corresponding to B_i and with $g_{i,M}$ the output bit corresponding to the monomial M. With these notations we have that

$$g_{i,M}(x_1, \ldots, x_m) = M(B_i) = M(x_{s_0 \cdot (i-1)+1}, \ldots, x_{s_0 \cdot i}),$$

$$G_i = (g_{i,M})_{M \in \mathcal{M}_k},$$

$$G^{(h)} = (G_1, \ldots, G_{i_0}) = (g_{i,M})_{i=1,\ldots,i_0, M \in \mathcal{M}_k}.$$

The length of the output of $G^{(h)}$ is clearly $i_0\binom{s_0}{k} = n$. Note that we ignore the input bits that fell in the leftover set.

For example, let $m = 7$, $\gamma = 1/2$ and $k = 2$. Then $s_0 = 3$, $i_0 = 2$, $\mathcal{M}_3 = \{y_1y_2, y_1y_3, y_2y_3\}$, $B_1 = \{x_1, x_2, x_3\}$ and $B_2 = \{x_4, x_5, x_6\}$. We get that $G_1 = (x_1x_2, x_1x_3, x_2x_3)$, $G_2 = (x_4x_5, x_4x_6, x_5x_6)$, and $G^{(h)} = (x_1x_2, x_1x_3, x_2x_3, x_4x_5, x_4x_6, x_5x_6)$.

We now show that for every $w \in \{0,1\}^n$ such that $\mathrm{wt}(w) \geq an$ we have that $\mathrm{bias}_w(G_h) \leq \epsilon$. Indeed let w be such that $\mathrm{wt}(w) \geq an$. For convenience we enumerate the coordinates of w in the same way as the coordinates of $G^{(h)}$, that is $w = (w_{i,M})_{i=1,\ldots,i_0, M \in \mathcal{M}_k}$. We also partition w to i_0 disjoint sets $w = (w_1, \ldots, w_{i_0})$, where $w_i = (w_{i,M})_{M \in \mathcal{M}_k}$. We note that as $\mathrm{wt}(w) \geq \delta n$ then the supports of at least $\lceil a \cdot \frac{n}{\binom{s_0}{i_0}} \rceil = \lceil a \cdot i_0 \rceil$ of the w_i's are not empty. We now have that,

$$\langle w, G^{(h)} \rangle = \bigoplus_{i=1}^{i_0} \langle w_i, G_i \rangle = \bigoplus_{i=1}^{i_0} \left(\sum_{M \in \mathcal{M}_k} w_{i,M} g_{i,M} \right) = \bigoplus_{i=1}^{i_0} p_i(B_i),$$

where each $p_i(B_i)$ is a degree k polynomial, over GF(2), in the variables of B_i. Denote with I the set of indices for which $p_i \neq 0$. As each p_i is a sum of different degree k monomials we have that the size of I is equal to the number of non empty w_i-s. Thus,

$$|I| \geq \lceil a \cdot i_0 \rceil \geq a \cdot i_0 = a \left\lfloor \frac{m}{\lfloor \gamma m \rfloor} \right\rfloor \geq a \left(\frac{m}{\lfloor \gamma m \rfloor} - 1 \right) \geq a \left(\frac{1}{\gamma} - 1 \right) > 2^{k-1} \log(1/2\epsilon).$$

As the sets B_i are disjoint the polynomials $p_i(B_i)$ for $i \in I$, viewed as random variables in the input bits, are independent random variables. By the Schwartz-Zippel lemma (Lemma 9), we get that the bias of each p_i, for $i \in I$ is at most $\frac{1}{2} - \frac{1}{2^k}$, and so by Lemma 7 we get that

$$\mathrm{bias}_w(G_h) \leq \frac{1}{2} \left(1 - \frac{2}{2^k} \right)^{|I|} < \frac{1}{2} \left(1 - \frac{1}{2^{k-1}} \right)^{2^{k-1} \log(1/2\epsilon)} \leq \epsilon.$$

When $n < i_0\binom{s_0}{k}$ we make a small modification to the construction. Let $t_0 = \lfloor \frac{n}{i_0} \rfloor$. Let $i_1 = n - i_0 \cdot t_0$. Clearly $i_1 < i_0$. For every set in the partition we shall define t_0 output bits, and for the first i_1 sets we shall define an additional output bit. The total number of output bits is $i_0 \cdot t_0 + i_1 = n$ as required. Instead of computing the value of every monomial of $\mathcal{M}_k$ on every set of the partition, we only output the value of the first t_0+1 monomials (according to, say, the lexicographical ordering of their exponent vector) on the first i_1 sets of the partition, and the value of the first t_0 monomials on the rest of the sets of the partition (i.e. on the remaining $i_0 - i_1$ sets). In this way the output length is $i_1(t_0+1) + (i_0 - i_1)t_0 = i_0 \cdot t_0 + i_1 = n$. The analysis of the bias also requires a small modification. As before we consider the partition of w to disjoint sets, where the first i_1 sets are of size t_0+1 and the last $i_0 - i_1$ sets are of size t_0. We define p_i and I as before. We have that

$$|I| \geq \left\lceil \frac{an}{t_0+1} \right\rceil \geq \frac{an}{\lfloor \frac{n}{i_0} \rfloor + 1} \geq^{(*)} a(i_0 - 1) \geq a(\frac{1}{\gamma} - 2) = 2^{k-1} \log(1/2\epsilon), \tag{1}$$

where inequality $(*)$ follows from the lower bounds on ϵ, and n (for completeness we give the proof of $(*)$ in Appendix A). We get that

$$\text{bias}_w(G_h) \leq \frac{1}{2}\left(1 - \frac{2}{2^k}\right)^{|I|} \leq \frac{1}{2}\left(1 - \frac{1}{2^{k-1}}\right)^{2^{k-1}\log(1/2\epsilon)} \leq \epsilon.$$

It is clear that the complexity of computing this encoding is polynomial in n. This completes the proof of the Theorem. □

3.2 From heavy to light

In this section we prove a theorem that shows that in order to construct an ϵ-biased generator it is sufficient to construct a generator whose output is almost unbiased w.r.t. "heavy" tests. The basic tool in proving this theorem is a linear transformation from $\{0,1\}^n$ to itself, that sends all the non-zero vectors in the Hamming ball of radius an (light vectors) to vectors of weight at least bn (heavy vectors). Stated differently, this linear transformation has the property that it takes any two vectors that are at distance at most an and sends them to vectors at distance at least bn. This definition immediately brings to mind error correcting codes, and indeed the construction of such transformations is based on the generating matrix of a suitable error correcting code.

Definition 11 *A linear transformation $A : \{0,1\}^n \to \{0,1\}^n$ is* (a,b)-expanding *if for every $v \in \{0,1\}^n$ such that $\text{wt}(v) \leq an$ we have that $\text{wt}(v^t A) = \text{wt}(A^t v) \geq bn$. We say that A is* symmetric (a,b)-expanding *if in addition for every $u \in \{0,1\}^n$ such that $\text{wt}(u) \leq an$ we have that $\text{wt}(Au) \geq bn$.*

We now show how to construct symmetric expanding linear transformations.

Theorem 12 *(Expanding transformations) Assume that there exists an explicit construction of a linear (self-dual) $[\frac{1}{2}, \delta]$-code of block length $2n$ over* GF(2)*. Then there is an explicit (symmetric) $(a, \delta - a)$-expanding transformation $A : \{0,1\}^n \to \{0,1\}^n$ that can be constructed in the same time (up to an additive $O(n^3)$ term) as the generating matrix of the underlying code.*

Proof Let C be a $[\frac{1}{2}, \delta]$ code of block length $2n$ (in particular the rate of C is n). Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be the generating matrix of C. As the rate is $1/2$ we can assume w.l.o.g. that G has the following form $G = \begin{pmatrix} I \\ A \end{pmatrix}$ where I is the $n \times n$ identity matrix and A is an $n \times n$ matrix. Let $\vec{0} \neq w \in \{0,1\}^n$ be a vector of weight $\leq an$. Then $\delta n \leq \mathrm{wt}(Gw) = \mathrm{wt}(w) + \mathrm{wt}(Aw) \leq an + \mathrm{wt}(Aw)$. In particular $\mathrm{wt}(Aw) \geq \delta n - an$. Thus the matrix A^t is $(a, \delta - a)$-expanding.

Let us now assume that C is also self-dual. It is easy to see that the matrix $H = \begin{pmatrix} A^t \\ I \end{pmatrix}$ is a generating matrix for the dual code, and as C is self-dual we have that H is also a generating matrix for C. As before we get that $\delta n \leq \mathrm{wt}(Hw) = \mathrm{wt}(A^t w) + \mathrm{wt}(w) \leq an + \mathrm{wt}(A^t w)$. Hence, $\mathrm{wt}(A^t w) \geq \delta n - an$. Together with the above observation that $\mathrm{wt}(Aw) \geq \delta n - an$ we get that A is symmetric $(a, \delta - a)$-expanding. □

By applying Theorem 12 on the codes obtained from Theorem 3 we get the following corollary.

Corollary 13 *For every $0 < a < 0.0595$ and every large enough (independent of a) integer t there is an explicit symmetric $(a, 0.0595 - a)$-expanding transformation of dimension $189 \cdot 8^t$ that can be constructed in time polynomial in its dimension.*

We now show that using expanding linear transformations we can transform any generator that is ϵ-biased w.r.t. heavy tests to an ϵ-biased generator.

Theorem 14 *Let A be an (a, a)-expanding transformation of dimension n. Let $G^{(h)} : \{0,1\}^m \to \{0,1\}^n$ satisfy that there exists an $0 < \epsilon$ such that for every $w \in \{0,1\}^n$ with $\mathrm{wt}(w) \geq an$, $\mathrm{bias}_w(G^{(h)}) \leq \epsilon$. Define $G^{(l)} : \{0,1\}^m \to \{0,1\}^n$ in the following way $G^{(l)}(x) = A(G^{(h)}(x))$. Then for every $w \in \{0,1\}^n$ with $\mathrm{wt}(w) \leq an$, $\mathrm{bias}_w\left(G^{(l)}\right) \leq \epsilon$. In particular, if we define $G : \{0,1\}^{2m} \to \{0,1\}^n$ as*

$$G(x, y) = G^{(h)}(x) \oplus G^{(l)}(y)$$

we get that $\mathrm{bias}(G) \leq \epsilon$. The map $G^{(l)}$ can be constructed in time polynomial in the construction time of $G^{(h)}$ and of A, and hence so can G.

Proof Let $w \in \{0,1\}^n$ be such that $\mathrm{wt}(w) \leq an$. It is clear that

$$\mathrm{bias}_w\left(G^{(l)}\right) = \mathrm{bias}_w\left(A(G^{(h)}(x))\right) = \left|\frac{1}{2} - \mathbb{E}_{x \in \{0,1\}^m}\langle A\left(G^{(h)}\right), w\rangle\right|$$

$$= \left| \frac{1}{2} - \mathbb{E}_{x \in \{0,1\}^m} \langle G^{(h)}, A^t w \rangle \right| = \mathrm{bias}_{A^t w} \left(G^{(h)} \right).$$

As A is (a,a)-expanding we have that $\mathrm{wt}(A^t w) \geq an$. By the assumption on $G^{(h)}$ we get that $\mathrm{bias}_{A^t w}(G^{(h)}) \leq \epsilon$ and so $\mathrm{bias}_w \left(G^{(l)} \right) \leq \epsilon$. The claim regarding the bias of G is an immediate corollary of Lemma 6. The claim regarding the construction time of G is obvious. □

We note that as A is a linear transformation then the degree of G (i.e. the maximal degree of its output bits viewed as polynomials, over GF(2)), in the input variables is at most the degree of $G^{(h)}$.

3.3 Proof of Theorem 1

The proof of Theorem 1 follows from Theorem 10, Corollary 13 and Theorem 14. Indeed, let $a = 0.0595/2$. Given two integers $0 < k, m$ and $\epsilon \geq \exp(-O(\frac{am^{1-\frac{1}{k}}}{2^k k}))$, let t be defined as the largest integer such that $189 \cdot 8^t \leq i_0 \binom{s_0}{k}$, where s_0, i_0 are as in Theorem 10. We want m to be large enough so that Theorem 3 will ensure the existence of $[\frac{1}{2}, 2a]$-codes of block length $\frac{189}{4} \cdot 8^{t+1}$. Let $n = 189 \cdot 8^t$. Clearly $\frac{i_0}{8} \binom{s_0}{k} < n \leq i_0 \binom{s_0}{k}$. Using the notations of Theorem 10 we bound n from below by

$$n > \frac{1}{8} i_0 \binom{s_0}{k} > \left(\frac{\gamma m}{k} \right)^k = \Omega \left(\left(\frac{m}{2^k k \log(1/\epsilon)} \right)^k \right).$$

By Corollary 13 we can construct in polynomial time an (a,a)-expanding transformation of dimension n. For our k, m, n, a, ϵ let $G^{(h)}$ be the generator obtained from theorem 10. Clearly $G^{(h)}$ satisfies the conditions of Theorem 14. Let G be the generator obtained from Theorem 14. It is easy to verify that G is the desired generator. □

4 Construction of ϵ-biased good codes

The construction of ϵ-biased good codes follows the same lines as the construction of low-degree ϵ-biased generators. The main difference is that we don't have to keep the degree low but rather make sure that the generator outputs a good code. Thus, we will need a generator for heavy tests that outputs a good code and a way of transforming this generator in to a truly unbiased generator that also outputs a good code. The difference from the proof of Theorem 1 is in two points. First we will need a different construction of a generator $G^{(h)}$ for heavy tests (compare to Theorem 10). Then we will need a construction of a *symmetric* expanding transformation (see Definition 11) that will enable us to transform this generator to an ϵ-biased good code (compare to Theorem 14).

As before we start by giving a construction of a good code that is also unbiased w.r.t. heavy tests. We then show a general way of transforming codes that are unbiased w.r.t. heavy tests to ϵ-biased codes.

4.1 Construction of codes that are unbiased w.r.t. heavy tests

Theorem 15 *Let $0 < a < 1$ be a constant. Let n be an integer. For $an/48 \leq M < an/6$ let $\hat{C}$ be an $[M, m, d]$ binary* linear *error correcting code that has a polynomial time encoding algorithm (i.e. its generating matrix can be computed in polynomial time) and a polynomial time decoding algorithm that can correct $\hat{\alpha} > 0$ fraction of errors. Denote $\hat{R} = m/M$ and $\hat{\delta} = d/M$. Then there is a polynomial time constructible map $G^{(h)} : \{0,1\}^m \to \{0,1\}^n$ such that its image, $C = G^{(h)}(\{0,1\}^m)$ has the following properties.*

- *C is an $[R, \delta]$-code of block length n for $R \geq \frac{a}{48}\hat{R}$ and $\delta \geq \frac{a}{48}\hat{\delta}$.*
- *The weight of every $v \in C$ is bounded from above by $\mathrm{wt}(v) \leq an/3$.*
- *C has a polynomial time encoding algorithm and a polynomial time decoding algorithm that can correct from $\alpha \geq \frac{a}{48}\hat{\alpha}$ fraction of errors.*
- *For every $w \in \{0,1\}^n$ such that $\mathrm{wt}(w) \geq an$ we have that $\mathrm{bias}_w(C) \leq \exp(-O(n))$.*

Proof We start with a sketch of the proof. The proof is similar in nature to the proof of Theorem 10. As before we partition the input string to roughly m/k subsets of size k, for some k. For each of the subsets we define 2^{k-2} distinct output bits that correspond to the values of 2^{k-2} linearly independent polynomials, in k variables, evaluated on the input bits that belong to the subset. In this way we get roughly $2^{k-2}m/k$ output bits. We then concatenate (in the sense of string concatenation) to these output bits the encoding of the input bits w.r.t. the code $\hat{C}$. This defines the map $G^{(h)}$. It remains to show that $G^{(h)}$ has the required properties. Indeed, the first $2^{k-2}m/k$ output bits will assure us that the weight of each output word is not too large and that (as in the proof of Theorem 10) the output has a small bias. The concatenation of $\hat{C}$ ensures that the distance between any two output words is linearly large which gives us the required decoding property. As before, in order to be completely accurate we have to handle the case that for our k (that will be later specified) m/k is not an integer, and the case where $2^{k-2}m/k > n - M$. We now give the formal proof.

Let k be the smallest integer satisfying $\lfloor m/k \rfloor 2^{k-2} \geq n - M$. It is clear that k is a constant depending only on a and $\hat{R}$. Let $i_0 = \lfloor m/k \rfloor$, $t_0 = \lfloor (n-M)/i_0 \rfloor$ and $i_1 = n - M - i_0 \cdot t_0$. Clearly $i_1 < i_0$. Let $x = (x_1, \ldots, x_m)$ be our input. Partition the first $k\lfloor m/k \rfloor = ki_0$ bits to i_0 sets of size k. The i-th set in the partition is $B_i = \{x_{(i-1)k+1}, \ldots, x_{ik}\}$. For each of the B_i's we define t_0 output bits, and for the first i_1 sets we define an additional output bit. This gives a total of $i_0 t_0 + i_1 = n - M$ output bits.

Denote with $\{\chi_0, \ldots, \chi_{2^{k-2}-1}\}$ the following characteristic functions in k variables:

$$\chi_j(y_1, \ldots, y_k) = 1 \Leftrightarrow (y_1 = 1) \wedge (y_2 = 1) \wedge \left(\sum_{i=3}^{k} y_i 2^{i-1} = j \right).$$

Equivalently, let $j_0, \ldots, j_{k-3}$ be the binary representation of j when j_0 is the LSB and j_{k-3} is the MSB (i.e. $j = \sum_{i=0}^{k-3} j_i 2^i$). We have that

$$\chi_j(y_1, \ldots, y_k) = y_1 \cdot y_2 \cdot \prod_{i=3}^{k} (y_i - j_{i-3} + 1) \mod 2.$$

It is easy to see that the χ_j-s are linearly independent, and that on every input at most one of the χ_j-s is non-zero. Note that the degree of every monomial of χ_j is at least 2 and at most k. We denote with G_i the set of output bits corresponding to B_i and with $g_{i,j} \in G_i$ the output bit corresponding to χ_j. Namely,

$$g_{i,j}(x_1, \ldots, x_m) = \chi_j(x_{k\cdot(i-1)+1}, \ldots, x_{k\cdot i}).$$

With these notations we have that

$$\forall 1 \leq i \leq i_1 \;\; G_i = (g_{i,j})_{j=0,\ldots,t_0},$$

$$\forall i_1 < i \leq i_0 \;\; G_i = (g_{i,j})_{j=0,\ldots,t_0-1}.$$

The length of the output is clearly $i_1(t_0 + 1) + (i_0 - i_1)t_0 = n - M$. Denote with $G_{\hat{C}} : \{0,1\}^m \to \{0,1\}^M$ the generating matrix of the code $\hat{C}$. In particular, the encoding of the vector $x = (x_1, \ldots, x_m)$ is $G_{\hat{C}} \cdot x$. We now define the map $G^{(h)}$:

$$G^{(h)} = (G_1, \ldots, G_{i_0}, G_{\hat{C}}),$$

that is, on an input x we first have $n - M$ output bits that come from $G_1, \ldots, G_{i_0}$, and the last M bits are the encoding of x w.r.t. to the code $\hat{C}$. Clearly the output length is n. Let $C = G^{(h)}(\{0,1\}^m)$ be the image of $G^{(h)}$. We show that C has the required properties.

The rate of C is m and thus its relative rate R is

$$R = m/n \geq m/(48M/a) = \frac{a}{48}\hat{R}.$$

It is clear that C contains $\hat{C}$ as its last M bits and so the minimal distance of C is at least $\hat{\delta}M$. Thus the relative minimal distance of C is:

$$\delta \geq \hat{\delta}M/n \geq \frac{a}{48}\hat{\delta}.$$

In order to bound the weight of every $v \in C$, we recall that in every G_i at most one of the output bits is non-zero. Thus the total weight of $v \in C$ is bounded from above by $i_0 + M$. We get that

$$\forall v \in C, \;\; \mathrm{wt}(v) \leq i_0 + M = \lfloor m/k \rfloor + M \leq M/k + M \leq 2M \leq an/3.$$

To show the error correcting property we note that if the total number of errors is $\hat{\alpha}M$, then in particular the last M bits of C contain at most $\hat{\alpha}M$ errors. As the last M bits of C correspond to a codeword in $\hat{C}$ we can use the decoding algorithm of $\hat{C}$ to obtain the original message. Thus we can fix at least $\hat{\alpha}M \geq \frac{a}{48}\hat{\alpha}n$ errors.

It remains to show that C is ϵ-biased w.r.t. words of weight $\geq an$. Indeed let $\mathrm{wt}(w) \geq an$. As in the proof of Theorem 10 we partition w to $i_0 + 1$ disjoint sets $w = (w_1, \ldots, w_{i_0}, w_{\hat{C}})$, where the number of bits in w_i is the same as the number of output bits in G_i. We also write $w_i = (w_{i,0}, \ldots, w_{i,|G_i|-1})$. Since $\mathrm{wt}(w) \geq an$ we have that the supports of at least

$$\left\lceil \frac{an - M}{2^{k-2}} \right\rceil \geq \left\lceil \frac{a - a/6}{2^{k-2}} n \right\rceil$$

of $w_1, \ldots, w_{i_0}$ are not empty. As in Theorem 10 we get that

$$\langle w, G^{(h)} \rangle = \left(\bigoplus_{i=1}^{i_0} \langle w_i, G_i \rangle \right) \oplus \langle w_{\hat{C}}, G_{\hat{C}} \rangle = \left(\bigoplus_{i=1}^{i_0} \left(\sum_{g_{i,j} \in G_i} w_{i,j} \chi_{i,j}(B_i) \right) \right) \oplus \langle w_{\hat{C}}, G_{\hat{C}} \rangle$$

$$= \left(\bigoplus_{i=1}^{i_0} p_i(B_i) \right) \oplus \langle w_{\hat{C}}, G_{\hat{C}} \rangle,$$

where $p_i(B_i)$ is a polynomial over GF(2) in the variables of B_i. As $G_{\hat{C}}$ is a linear function in $\{x_1, \ldots, x_m\}$ we have that

$$\langle w_{\hat{C}}, G_{\hat{C}} \rangle = \sum_{i=1}^{i_0} \ell_i(B_i),$$

where the ℓ_i's are linear functions. We thus have that

$$\langle w, G^{(h)} \rangle = \left(\bigoplus_{i=1}^{i_0} p_i(B_i) \right) \oplus \left(\sum_{i=1}^{i_0} \ell_i(B_i) \right) = \left(\bigoplus_{i=1}^{i_0} \tilde{p}_i(B_i) \right),$$

where each $\tilde{p}_i$ is a polynomial over GF(2) in the variables of B_i.

Denote with I the set of indices for which $p_i \neq 0$. As each p_i is a sum of linearly independent polynomials we have that the size of I is equal to the number of non empty w_i-s. Since $\tilde{p}_i = p_i + \ell_i$ and each monomial of p_i has degree at least 2 we get that if $p_i \neq 0$ then $\tilde{p}_i \neq 0$ (because p_i and ℓ_i cannot cancel each other). We conclude that at least $\left\lceil \frac{a - a/6}{2^{k-2}} n \right\rceil$ of the p_i's are non-zero polynomials (of degree at most k). As the sets B_i are disjoint the polynomials $\tilde{p}_i(B_i)$ for $i \in I$, viewed as random variables in the input bits, are independent random variables. By the Schwartz-Zippel lemma (Lemma 9), we get that the bias of each p_i, for $i \in I$ is at most $\frac{1}{2} - \frac{1}{2^k}$, and so by Lemma 7 we get that

$$\mathrm{bias}_w(G_h) \leq \frac{1}{2} \left(1 - \frac{2}{2^k} \right)^{|I|} \leq \frac{1}{2} \left(1 - \frac{1}{2^{k-1}} \right)^{\left\lceil \frac{a - a/6}{2^{k-2}} n \right\rceil} = \exp\left(-O\left(\frac{a}{2^{2k}} n \right) \right) = \exp\left(-O\left(n \right) \right).$$

□

4.2 From heavy to light: keeping the distance large

For the purpose of constructing ϵ-biased codes we shall need the more powerful notion of a symmetric expanding transformation. We will also require that the transformation has an efficient decoding algorithm (in some special sense).

Definition 16 *An (a,b)-expanding transformation A of dimension n can* decode *from α fraction of errors if there exists a decoding algorithm D such that for any two vectors $v, err \in B^n$, satisfying $\mathrm{wt}(v) \leq an$ and $\mathrm{wt}(err) \leq \alpha n$, we have that $D(Av + err) = v$.*

Theorem 17 *(Symmetric expanding transformations) For any $[\frac{1}{2}, \delta]$ self-dual code C of block length $2n$, that has a polynomial time decoding algorithm that can handle α fraction of errors, and for every $0 < a < \delta$, there is an explicit symmetric $(a, \delta - a)$-expanding transformation A of dimension n, that has a polynomial time algorithm for decoding from $2\alpha - a$ fraction of errors. Moreover, the constructing time of A is the same (up to $\pm O(n^3)$) as the time for constructing the generating matrix of the code, and the running time of the decoding algorithm is the same (up to $\pm O(n)$) as the running time of the decoding algorithm of C.*

Proof Let G be the generating matrix of C. W.l.o.g we can assume that $G = \begin{pmatrix} I \\ A \end{pmatrix}$. The proof of Theorem 12 shows that A is symmetric $(a, \delta - a)$-expanding. Thus, it remains to prove the decoding property. Given a vector of the form $A(v) + err$, where $\mathrm{wt}(v) \leq an$, and $\mathrm{wt}(err) \leq (2\alpha - a)n$, consider the $2n$ dimensional vector $(\vec{0}, A(v) + err)$, where $\vec{0}$ is the n-th dimensional zero vector. The distance of this vector from the vector $(v, A(v))$ is at most $an + (2\alpha - a)n = 2\alpha n$. As (v, Av) belongs to the image of G (and hence belongs to C) we can apply the given decoding algorithm for C (that can correct $\alpha \cdot 2n$ errors) on the word $(\vec{0}, A(v) + err)$ to get the word (v, Av) from which we get v. The claim regarding the running time is obvious. □

Applying Theorem 17 on the codes obtained from Theorem 3 we get the following corollary.

Corollary 18 *For every $0 < a < 0.0595$ and every large enough integer t there is an explicit symmetric $(a, 0.0595 - a)$-expanding transformation A, of dimension $n = 189 \cdot 8^t$, that can be constructed in polynomial time (in n), and that has a polynomial time decoding algorithm that can correct $0.0595 - a$ fraction of errors.*

We now show that by using symmetric expanding transformations we can transform a good code that is ϵ-biased w.r.t. heavy tests to a good code that is ϵ-biased.

Theorem 19 *Let A be a symmetric (a,a)-expanding transformation of dimension n that can correct β fraction of errors. Let $G^{(h)} : \{0,1\}^m \to \{0,1\}^n$ be a mapping whose image is a code $C \subset \{0,1\}^n$ with the following properties.*

- *C is an $[R, \delta]$ code of block length n.*
- *For every word $v \in C$ we have that $\mathrm{wt}(v) \leq \Delta n < an/2$.*

- *C has a polynomial time decoding algorithm that can correct α fraction of errors.*
- *For every $w \in \{0,1\}^n$ such that* $\mathrm{wt}(w) \geq an$, $\mathrm{bias}_w(C) \leq \epsilon$.

Let $G^{(l)}(x) = A(G^{(h)}(x))$. Finally let $G : \{0,1\}^{2m} \to \{0,1\}^n$ be the following generator $G(x,y) = G^{(h)}(x) \oplus G^{(l)}(y)$. Let $\tilde{C} \subset \{0,1\}^n$ be the image of G (note that as sets we have that $\tilde{C} = C \oplus A(C)$). Then $\tilde{C}$ has the following properties.

- *$\tilde{C}$ has relative rate $2R$ and relative minimal distance at least* $\min(\delta, a - 2\Delta)$.
- *$\tilde{C}$ has a polynomial time decoding algorithm that can correct* $\min(\beta - \Delta, \alpha)$ *fraction of errors.*
- $\mathrm{bias}\left(\tilde{C}\right) \leq \epsilon$ *(and hence* $\mathrm{bias}(G) \leq \epsilon$*).*

Proof We prove the properties of $\tilde{C}$ one by one. It is clear that the relative rate of $\tilde{C}$ is $2R$. Let v, u be two different code words in $\tilde{C}$. Then we can write $v = v_1 + Av_2$ and $u = u_1 + Au_2$ for $v_1, v_2, u_1, u_2 \in C$. It is clear that $\mathrm{dist}(u,v) = \mathrm{dist}(u_1 + Au_2, v_1 + Av_2) = \mathrm{dist}(u_1 - v_1, A(v_2 - u_2))$. As $u_1, v_1, u_2, v_2 \in C$ we have that $\mathrm{wt}(u_1 - v_1) \leq \mathrm{wt}(u_1) + \mathrm{wt}(v_1) \leq 2\Delta n < an$ and similarly that $wt(v_2 - u_2) \leq 2\Delta n < an$. We analyze two cases.
Case $v_2 \neq u_2$: As A is symmetric (a,a)-expanding we have that $\mathrm{wt}(A(v_2 - u_2)) \geq an$. Thus, $\mathrm{dist}(u_1 - v_1, A(v_2 - u_2)) \geq (a - 2\Delta)n$ (this is a trivial bound on the distance of a vector of weight at most 2Δ and a vector of weight at least a). It follows that $\mathrm{dist}(u,v) \geq (a - 2\Delta)n$.
Case $v_2 = u_2$: In this case $\mathrm{dist}(u,v) = \mathrm{dist}(u_1, v_1) \geq \delta n$, as C has minimal distance $\geq \delta n$ (note that we must have that $v_1 \neq u_1$).
Combining the two cases we get that the relative distance of $\tilde{C}$ is at least $\min(a - 2\Delta, \delta)$.

We now show a decoding algorithm for $\tilde{C}$. Let u be a codeword in $\tilde{C}$, and let $err \in \{0,1\}^n$ be an error vector of weight $\mathrm{wt}(err) \leq \min(\alpha, \beta - \Delta) \cdot n$. Let $v_1, v_2 \in C$ be such that $u = v_1 + Av_2$. We now show how to recover v_1, v_2 from the corrupted word $u + err$. Let $\widetilde{err} = err + v_1$. As $\mathrm{wt}(err) \leq (\beta - \Delta)n$ and $\mathrm{wt}(v_1) \leq \Delta n$ we have that $\mathrm{wt}(\widetilde{err}) \leq \beta n$. By our assumption A can decode from β fraction of errors, thus A can recover the value of v_2 from the input $Av_2 + \widetilde{err}$. As $u + err = Av_2 + (v_1 + err) = Av_2 + \widetilde{err}$, we can recover v_2 from the input $u + err$. Given v_2 and the word $u + err$ we can get the vector $v_1 + err$ as $v_1 + err = u + err - Av_2$. Since $\mathrm{wt}(err) \leq \alpha n$ we can use the decoding algorithm of C to get v_1 from the input $v_1 + err$. Clearly the running time of this decoding algorithm is polynomial whenever the running time of the decoding algorithms for A and C are polynomial. This shows the decoding property.

Finally we notice that as a direct consequence of Theorem 14 we get that G is an ϵ-biased generator (equivalently, that $\tilde{C}$ is an ϵ-biased set).

□

4.3 Proof of Theorem 2

In order to prove Theorem 2, we apply Theorem 15 on the codes promised by Theorem 3, to get a good code that is unbiased w.r.t. heavy tests. Then we apply Theorem 19 on this code to obtain a good code that is ϵ-biased. Details follow.

Let $a = 0.0595/2$. Given a positive integer $n = 189 \cdot 8^t$, let s be the largest integer such that $\frac{189}{4}8^{s+1} \leq an/6$. Let $m = 189 \cdot 8^s$, then clearly $m > an/96$. We want t (and so n) to be large enough so that Theorem 3 will guarantee the existence of a $[\frac{1}{2}, 2a]$-code of block length $\frac{189}{4}8^{s+1}$, that we denote $\hat{C}$. In other words, $\hat{C}$ is a self-dual $[\frac{1}{2}, 2a]$-code of rate m and block length $2m$. Recall that $\hat{C}$ has a decoding algorithm that can correct at least $\hat{\alpha} \stackrel{\Delta}{=} a$ fraction of errors. By applying Theorem 15 on the code $\hat{C}$ we obtain a generator $G^{(h)} : \{0,1\}^m \to \{0,1\}^n$ that is (almost) unbiased against tests of weight at least an and whose image is a code C with the following parameters: The relative rate of C is $m/n \geq \frac{a}{96}$; the relative distance of C is at least $\frac{a}{48} \cdot 2a = \frac{a^2}{24}$; the weight of every codeword of C is at most $\frac{a}{3}n$; C has a decoding algorithm that can fix a fraction $\frac{a^2}{48}$ of errors; and the bias of C against words of weight at least an is at most $\epsilon = \exp(-O(n))$.

Now we apply Theorem 19 on the generator $G^{(h)}$, where A be the symmetric (a, a)-expanding transformation of dimension n guaranteed by Corollary 18. Recall that A has a decoding algorithm that can fix a fraction of a errors. We obtain a generator $G : \{0,1\}^{2m} \to \{0,1\}^n$ whose image is a code $\tilde{C}$ with the following parameters. The relative rate of $\tilde{C}$ is $2m/n > \frac{a}{48}$; the relative minimal distance of $\tilde{C}$ is at least $\min[a^2/24, a/3] = a^2/24$; $\tilde{C}$ has a decoding algorithm that can fix a $\min(2a/3, a^2/48) = a^2/48$ fraction of errors; and the bias of $\tilde{C}$ (and hence of G) is at most $\epsilon = \exp(-O(n/a))$. As the constructions of A and of $\hat{C}$, as well as their decoding algorithms, run in polynomial time, we get that C and hence $\tilde{C}$ have polynomial time encoding and decoding algorithms. This completes the proof of Theorem 2. □

Acknowledgements

I would like to thank Adam Smith for introducing me to the problem of constructing ϵ-biased good error correcting codes and for many valuable discussions. Thanks also to Oded Goldreich, Sofya Raskhodnikova and Avi Wigderson for helpful discussions and to Vinay Deolalikar, Simon Litsyn, Oded Regev, Ronny Roth and Sergey Yekhanin for helpful information regarding algebraic geometry error correcting codes. Finally I would like to thank Simon Litsyn for bringing [FR93] and [Mat02] to my attention.

References

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple construction of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.

[AIK+90] M. Ajtai, H. Iwaniec, J. Komlos, J. Pintz, and E. Szemeredi. Construction of a thin set with small fourier coefficients. *Bulletin of the London Mathematical Society*, 22:583–590, 1990.

[AIK04] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. In *45th annual FOCS*, pages 166–175, 2004.

[ALT01] A. Ashikhmin, S. Litsyn, and M. A. Tsfasman. Asymptotically good quantum codes. *Physical Review A*, 63:0322311, 2001.

[AM95] N. Alon and Y. Mansour. epsilon-discrepancy sets and their application for interpolation of sparse polynomials. *Information Processing Letters*, 54(6):337–342, 1995.

[AR94] N. Alon and Y. Roichman. Random cayley graphs and expanders. *Random Structures and Algorithms*, 5(2):271–285, 1994.

[BSSVW03] E. Ben-Sasson, M. Sudan, S. Vadhan, and A. Wigderson. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proceedings of the 35th STOC*, pages 612–621, 2003.

[CM01] M. Cryan and P. B. Miltersen. On pseudorandom generators in NC0. In *Proceedings of MFCS*, 2001.

[DS05] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proceedings of the 37th Annual STOC*, pages 654–663, 2005.

[EGL+92] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Velickovic. Approximations of general independent distributions. In *24th annual STOC*, pages 10–16, 1992.

[Fei95] J. Feigenbaum. The use of coding theory in computational complexity. In *Different Aspects of Coding Theory, Proceedings of Symposia on Applied Mathematics*, pages 207–233, 1995.

[FR93] G. L. Feng and T. R. N. Rao. Decoding algebraic-geometric codes up to the designed minimum distance. *IEEE Transactions on Information Theory*, 39(1):37–45, 1993.

[Gol00a] O. Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(90), 2000.

[Gol00b] O. Goldreich. Short locally testable codes and proofs (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, 5(14), 2000.

[Gur01] V. Guruswami. *List decoding of error correcting codes*. PhD thesis, MIT, 2001.

[Hås87] J. Håstad. One-way permutations in nc0. *Infrmation Processing Letters*, 26(3):153–155, 1987.

[HPS93] J. Håstad, S. Phillips, and S. Safra. A well-characterized approximation problem. *Information Processing Letters*, 47(6):301–305, 1993.

[KL01] M. Krause and S. Lucks. On the minimal hardware complexity of pseudorandom function generators. In *18th Annual STACS*, pages 419–430, 2001.

[KSV02] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2002.

[Mat02] R. Matsumoto. Improvement of ashikhmin-litsyn-tsfasman bound for quantum codes. *IEEE Transactions on Information Theory*, 48(7):2122–2124, 2002.

[MNN94] R. Motwani, J. Naor, and M. Naor. The probabilistic method yields deterministic parallel algorithms. *JCSS*, 49(3):478–516, 1994.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes, Part II*. North-Holland, 1977.

[MST03] E. Mossel, A. Shpilka, and L. Trevisan. On ϵ-biased generators in NC^0. In *44th Annual FOCS*, pages 136–145, 2003.

[MW02] R. Meshulam and A. Wigderson. Expanders from symmetric codes. In *34th annual STOC*, pages 669–677, 2002.

[Nao92] M. Naor. Constructing Ramsey graphs from small probability spaces. Technical report, IBM Research Report RJ 8810, 1992.

[NC00] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge, 2000.

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

[Raz05] R. Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual STOC*, pages 11–20, 2005.

[RSW93] A. A. Razborov, E. Szemerédi, and A. Wigderson. Constructing small sets that are uniform in arithmetic progressions. *Combinatorics, Probability & Computing*, 2:513–518, 1993.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *JACM*, 27(4):701–717, 1980.

[Tre04] L. Trevisan. Some applications of coding theory in computational complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, (043), 2004.

[Zip79] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and algebraic computation*, pages 216–226. 1979.

A Missing details from the proof of Equation (1)

We first notice that from the definition of γ, ϵ it follows (after a short calculation) that

$$2k < \gamma m. \tag{2}$$

Turning back to inequality $(*)$ from Equation 1, we see that it is enough to show that

$$n \geq \left(\left\lfloor \frac{n}{i_0} \right\rfloor + 1 \right) (i_0 - 1) \,. \tag{3}$$

We have that

$$\left(\left\lfloor \frac{n}{i_0} \right\rfloor + 1 \right) (i_0 - 1) \leq \left(\frac{n}{i_0} + 1 \right) (i_0 - 1) = n - \frac{n}{i_0} + i_0 - 1.$$

Thus, Equation (3) will follow if we prove that

$$\frac{n}{i_0} \geq \frac{1}{8} \binom{\lfloor \gamma m \rfloor}{k} > i_0. \tag{4}$$

From equation (2) we get that

$$\frac{1}{8} \binom{\lfloor \gamma m \rfloor}{k} > \frac{1}{8} \left(\frac{\gamma m}{k} \right)^k \tag{5}$$

and

$$i_0 = \left\lfloor \frac{m}{\lfloor \gamma m \rfloor} \right\rfloor \leq \frac{m}{\lfloor \gamma m \rfloor} \leq \frac{m}{\gamma m - 1} \leq \frac{2}{\gamma}. \tag{6}$$

Thus, Equation (4) follow if we show that

$$\frac{1}{8} \left(\frac{\gamma m}{k} \right)^k > \frac{2}{\gamma}$$

which follows immediately from the definition of γ and ϵ.

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'