T U M

I N S T I T U T F Ü R I N F O R M A T I K

On the Average Sensitivity of Testing Square-Free Numbers

A. Bernasconi, C. Damm, I. Shparlinsky

TUM-I9908
März 99

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-03-I9908-100/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

On the Average Sensitivity of Testing Square-Free Numbers

Anna Bernasconi
Institut für Informatik
Technische Universität München, D-80290 München, Germany
bernasco@informatik.tu-muenchen.de

Carsten Damm*
Fachbereich für Informatik
Universität Trier, D-54286 Trier, Germany
damm@uni-trier.de

Igor Shparlinski †
School of Mathematics, Physics, Computing and Electronics
Macquarie University, NSW 2109, Australia
igor@mpce.mq.edu.au

Abstract

We study combinatorial complexity characteristics of a Boolean function related to a natural number theoretic problem. In particular we obtain a linear lower bound on the average sensitivity of the Boolean function deciding whether a given integer is square-free. This result allows us to derive a quadratic lower bound for the formula size complexity of testing square-free numbers and a linear lower bound on the average decision tree depth. We also obtain lower bounds on the degrees of exact and approximative polynomial representations of this function.

*Supported by DFG grant Me 1077/14-1.
†Supported in part by ARC grant A69700294.

1 Introduction

In view of the many applications in modern cryptology Boolean functions related to number theoretic problems are a natural object to study from the complexity viewpoint. Recently for such functions several complexity lower bounds have been obtained for representations like unbounded fan-in Boolean circuits, decision trees, and real polynomials (see [1, 3, 4, 8, 17, 18]). The two main ingredients of these papers are harmonic analysis and estimates based on number theoretic considerations.

In this paper we focus to a purely combinatorial complexity characteristic: the *average sensitivity* of Boolean functions. The sensitivity of a function f on input w is defined as the number of bits such that flipping one of them will change the value of the function; the sensitivity of f is the maximum of the sensitivity of f on input w over all strings w of a given length; finally, the average sensitivity of f is the average (taken with respect to the uniform distribution) of the sensitivity of f on input w over all w of a given length. These definitions are made precise below. The sensitivity is of interest because it can be used to obtain lower bounds for the CREW PRAM complexity of Boolean functions (see [9, 10, 16, 19]), that is the complexity on a *parallel random access machine* with an unlimited number of all-powerful processors, such that simultaneous reads of a single memory cell by several processors are permitted, but simultaneous writes are not. The average sensitivity is a finer characteristic of Boolean functions which has been studied in a number of papers, see [2, 5, 14].

Our main result consists in a linear lower bound on the average sensitivity of testing square-free numbers. More precisely, we consider the function g which decides whether a given $(n+1)$-bit odd integer is square-free, that is the function for which

$$g(x_1, \ldots, x_n) = \begin{cases} 1, & \text{if } 2x+1 \text{ is square-free,} \\ 0, & \text{if } 2x+1 \text{ is square-full,} \end{cases} \qquad (1)$$

where $x = x_1 \ldots x_n$ is the bit representation of x, $0 \le x \le 2^n - 1$ (if necessary we add several leading zeros), and prove that for g a linear lower bounds on the average sensitivity holds. This lower bound is derived by studying the distribution of odd square-free numbers with a fixed binary digit.

We then apply this estimate to derive new lower bounds on the formula size, on the average depth of a decision tree and on the degree of certain polynomial representations for g.

The linear bound on the average sensitivity of g also provides an alternative proof for the statement proved in [3, 4] that g does not belong to the class $\mathbf{AC}^0$. On the other hand, even a stronger result has recently been obtained in [1].

2 Basic Definitions

Let $\mathfrak{B}_n = \{0,1\}^n$ denote the n dimensional Boolean cube.

For a binary vector $a \in \mathfrak{B}_n$ we denote by $a^{(i)}$ the vector obtained from a by flipping its ith coordinate. Now we introduce the main combinatorial parameters of Boolean functions $f : \mathfrak{B}_n \to \{0,1\}$ considered in this paper.

The **sensitivity of f at input** $a \in \mathfrak{B}_n$ is the number

$$\sigma_a(f) = \sum_{i=1}^{n} \left| f(a) - f(a^{(i)}) \right| .$$

The **sensitivity** of f is defined as

$$\sigma(f) = \max_{x \in \mathfrak{B}_n} \sum_{i=1}^{n} \left| f(x) - f(x^{(i)}) \right| ,$$

and the **average sensitivity** of f is

$$s(f) = 2^{-n} \sum_{x \in \mathfrak{B}_n} \sum_{i=1}^{n} \left| f(x) - f(x^{(i)}) \right| .$$

Clearly, $s(f) \leq \sigma(f) \leq n$ for any f.

The average sensitivity of a function f can be equivalently defined as the sum of the **influences** of all variables on f, where the influence of x_i on f, denoted $I_i(f)$, is the probability that flipping the i-th variable of a random Boolean input will flip the output. In other words, $I_i(f)$ is a measure of how influential is the variable x_i in determining the outcome of f. Precisely we have

$$I_i(f) = 2^{-n} \sum_{x \in \mathfrak{B}_n} \left| f(x) - f(x^{(i)}) \right| ,$$

which immediately implies

$$s(f) = \sum_{i=1}^{n} I_i(f) . \tag{2}$$

Formulae are defined in the following recursive way: the variables x_1, x_2, $\ldots, x_n$ and their negations $\neg x_1$, $\neg x_2$, $\ldots, \neg x_n$ are formulae; if F_1, F_2 are formulae, so are $F_1 \wedge F_2$ and $F_1 \vee F_2$. The **size** of a formula F is the number of occurrences of variables in it. Notice that a formula can be equivalently defined as a Boolean circuit whose fan-out of gates is bounded by one.

A **decision tree** with input variables $x_1, \ldots, x_n$ is a rooted binary tree in which each inner node is labeled with a variable and the edges leaving the node are labeled 0 and 1, respectively. Further each leaf v of the tree is labeled with some value $\lambda(v) \in \{0, 1\}$.

A decision tree T in a natural way defines a Boolean function f_T. For an input assignment a the computation proceeds as follows: starting from the root, at each visited inner node a certain variable x_i is tested. The computation proceeds along the edge labeled a_i. Define $f_T(a) = \lambda(v)$, where v is the eventually reached leaf.

For a decision tree T we say input $a \in \mathfrak{B}_n$ exits in depth i (denoted $D_a(T) = i$), if during the computation of $f_T(a)$ exactly i edges are passed.

The **depth** of the tree is $D(T) = \max\{D_a(T) \mid a \in \mathfrak{B}_n\}$ and its **average depth** is

$$\overline{D}(T) = 2^{-n} \sum_{a \in \mathfrak{B}_n}^{n} D_a(T).$$

For a Boolean function f let $D(f)$ and $\overline{D}(f)$, respectively, denote the minimal depth and the minimal average depth, respectively, of any decision tree T with $f_T = f$. Clearly, $\overline{D}(f) \leq D(f) \leq n$ for any f.

Further we mention the following definitions from [15]: for a Boolean function $f : \mathfrak{B}_n \to \{0, 1\}$ let the **real degree** of f, denoted by $\Delta(f)$, be the degree of the unique multilinear real polynomial $P(X_1, \ldots, X_n)$ for which

$$f(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$

holds for every $(x_1, \ldots, x_n) \in \mathfrak{B}_n$. Here multilinearity means, that each variable appears with degree at most 1.

We also define the **real approximate degree** of f, denoted by $\delta(f)$, as the degree of a multilinear real polynomial $P(X_1, \ldots, X_n)$ for which

$$|f(x_1, \ldots, x_n) - P(x_1, \ldots, x_n)| \leq 1/3$$

holds for every $(x_1, \ldots, x_n) \in \mathfrak{B}_n$. Certainly, in all our results $1/3$ can be replaced by any constant $\gamma < 1/2$.

Clearly, $\delta(f) \leq \Delta(f) \leq n$ for any f.

Throughout the paper we identify integers and their bit representations. In particular, we write $f(x)$ and $x^{(i)}$ for n-bit integers x, assuming that these apply to their bit representations.

3 Relations between Formula Size, Decision Tree Depth, Polynomial Degree, and Average Sensitivity

Let f be a Boolean function on n variables, and $L(f)$ denote the number of occurrences of variables in the minimal-size formula that computes f. Following [13], it is possible to restate Khrapchenko's Theorem, which gives lower bounds on the size of Boolean formulae, as follows. For $A \subseteq f^{-1}(0)$ and $B \subseteq f^{-1}(1)$, we define the $|B| \times |A|$ matrix Q, with $q_{uv} = 1$ if the binary strings $u \in B$ and $v \in A$ differ in exactly one component; otherwise $q_{uv} = 0$. Then

$$L(f) \geq \frac{1}{|A||B|} \left(\sum_{u \in A,\ v \in B} q_{uv} \right)^2.$$

From this version of Khrapchenko's Theorem, we can easily derive a lower bound on the formula size in terms of the average sensitivity.

Lemma 1. *Let f be a Boolean function depending on n variables and let p denote the probability that f takes the value 1. Then*

$$L(f) \geq \frac{1}{4p(1-p)} s(f)^2.$$

Proof. Let $A = f^{-1}(0)$ and $B = f^{-1}(1)$. We obtain the desired lower bound by observing that

$$\sum_{u \in A,\ v \in B} q_{uv} = 2^{n-1} s(f),$$

$|A| = 2^n (1-p)$ and $|B| = 2^n\, p$. □

Notice that this bound on the formula size in terms of average sensitivity was essentially mentioned also in [2, 5].

Lemma 2. *Let f be a Boolean function. Then*

$$\overline{D}(f) \geq s(f).$$

Proof. Let a be an input assignment. The inequality $D_a(T) \geq \sigma_a(f)$ holds since otherwise some untested variable still could decide about the function value. Hence, $\mathbf{E}[D_a(T)] \geq \mathbf{E}[\sigma_a(f)] = s(f)$, where $\mathbf{E}$ denotes the expectation with respect to a uniformly distributed random input a. □

Essentially the same inequality has been proved in [7] using harmonic analysis of Boolean functions.

Finally we mention the following inequalities which are a combination of the identity (2) with Corollary 2.5 of [15] and a weaker version of Lemma 3.8 of [15], respectively.

Lemma 3. *Let f be a Boolean function. Then*

$$\Delta(f) \geq s(f) \qquad \text{and} \qquad \delta(f) \geq (s(f)/6)^{1/2}.$$

4 Distribution of Square-Free Numbers

First of all we need a result about the uniformity of distribution of odd square-free numbers with a fixed binary digit.

Let i be an integer, $1 \le i \le n$ and let $\mathcal{N}_i$ denote the set of integers x, $0 \le x \le 2^n - 1$ such that $2x + 1 \equiv 0 \pmod 9$ and $2x^{(i)} + 1$ is square free. Let M_i be the number of elements in $\mathcal{N}_i$.

Lemma 4. *For any i, $1 \le i \le n$, the bound*

$$M_i = \frac{1}{\pi^2} 2^n + O\left(2^{3n/4}\right)$$

holds.

Proof. It is easy to see that $2x + 1 \equiv 0 \pmod 9$ is equivalent to the condition $x \equiv 4 \pmod 9$. Let $T_i(d)$ be the number of integers x, $0 \le x \le 2^n - 1$, such that

$$x \equiv 4 \pmod 9 \qquad \text{and} \qquad 2x^{(i)} + 1 \equiv 0 \pmod{d^2}. \tag{3}$$

By applying the inclusion-exclusion principle we derive that

$$M_i = \sum_{\substack{1 \le d \le 2^{(n+1)/2} \\ d \equiv 1 \pmod 2}} \mu(d)\, T_i(d)\,,$$

where $\mu(d)$ is the Möbius function. We recall that $\mu(1) = 1$, $\mu(d) = 0$ if d is square-full and $\mu(d) = (-1)^{\nu(d)}$ otherwise, where $\nu(d)$ is the number of prime divisors of $d \ge 2$.

It is easy to see that $2x + 1$ and $2x^{(i)} + 1$ are relatively prime because they differ by a power of 2. Therefore $T_i(d) = 0$ if $3|d$.

Let us now estimate $T_i(d)$ for d with $\gcd(6, d) = 1$.

The flipping position splits the bits of x into two parts. Let us denote by $t = \max\{i - 1, n - i\}$ the length of the longest part. Then it is obvious that for any fixing of the ith binary digit and all digits of the shortest part we obtain that the system of congruences (3) can be replaced by 2^{n-t} systems of congruences, each of them of the form

$$2^s z + a \equiv 0 \pmod 9 \qquad \text{and} \qquad 2^s z + b \equiv 0 \pmod{d^2}$$

with $0 \le z \le 2^t - 1$, for some integers s, a and b. Since $\gcd(d,3) = 1$, applying the Chinese Remainder Theorem we see that these congruences define z uniquely modulo $9d^2$. Thus there are $2^t/9d^2 + O(1)$ such values of z in the interval $0 \le z \le 2^t - 1$.

Putting everything together we obtain

$$T_i(d) = 2^{n-t}\left(\frac{2^t}{9d^2} + O(1)\right) = \frac{2^n}{9d^2} + O(2^{n-t}).$$

From the inequality $t \ge (n-1)/2$ we conclude that

$$T_i(d) = \frac{2^n}{9d^2} + O(2^{n/2}). \tag{4}$$

It is also clear that

$$T_i(d) \le 2^n/d^2. \tag{5}$$

Let $K \ge 1$ be an integer. Using (4) for $d \le K$ and (5) for $d > K$, and taking into account that

$$\sum_{K<d\le 2^{(n+1)/2}} d^{-2} < \sum_{d=K+1}^{\infty} \frac{1}{d(d-1)} = \sum_{d=K+1}^{\infty} \left(\frac{1}{d-1} - \frac{1}{d}\right) = 1/K$$

we obtain

$$\begin{aligned} M_i &= 2^n \sum_{\substack{1\le d\le K \\ \gcd(6,d)=1}} \frac{\mu(d)}{9d^2} + O\left(2^{n/2} \sum_{1\le d\le K} 1 + 2^n \sum_{K<d\le 2^{(n+1)/2}} d^{-2}\right) \\ &= 2^n \sum_{\substack{1\le d\le K \\ \gcd(6,d)=1}} \frac{\mu(d)}{9d^2} + O\left(2^{n/2}K + 2^n K^{-1}\right). \end{aligned}$$

Extending the summation range in the first sum to all integers d introduces an additional error of order $2^n K^{-1}$. Therefore

$$M_i = 2^n \sum_{\gcd(6,d)=1} \frac{\mu(d)}{9d^2} + O\left(2^{n/2}K + 2^n K^{-1}\right) \tag{6}$$

for any integer K, $1 \le K \le 2^{(n+1)/2}$.

It is easy to verify that

$$\sum_{\gcd(6,d)=1} \frac{\mu(d)}{d^2} = \left(1 - \frac{1}{4}\right)^{-1} \left(1 - \frac{1}{9}\right)^{-1} \sum_{d=1}^{\infty} \frac{\mu(d)}{d^2} = \frac{3}{2} \sum_{d=1}^{\infty} \frac{\mu(d)}{d^2}.$$

From Theorem 287 of [11] we derive

$$\sum_{\gcd(6,d)=1} \frac{\mu(d)}{d^2} = \frac{9}{\pi^2}.$$

Selecting $K = \left\lfloor 2^{n/4} \right\rfloor$ in (6), we obtain the desired result. □

5 Average Sensitivity of Testing Square-Free Numbers

At this point we are able to derive our main result, namely a linear lower bound on the average sensitivity of testing square-free numbers.

Theorem 5. *For the Boolean function g given by (1) the bound*

$$s(g) \ge \frac{1}{\pi^2} n + o(n)$$

holds.

Proof. It is easy to see that, for any i, $1 \le i \le n$

$$I_i(f) \ge 2^{-n} M_i .$$

Since the average sensitivity is defined by the sum of the influences of all variables, applying Lemma 4 we obtain the desired estimate. □

We can now apply this estimate to derive non-trivial lower bounds for both the formula size and the average decision tree depth of the function (1).

Theorem 6. *For the Boolean function g given by (1) the bound*

$$L(g) \geq \frac{1}{32(\pi^2 - 8)} n^2 + o(n^2),$$

holds.

Proof. Let p be the probability that g takes the value 1. By applying the same elementary considerations we used in the proof of Lemma 4, it is easy to show that $p = 8\pi^{-2} + o(1)$. Combining Theorem 5 with Lemma 1 we then derive the desired statement. □

Using Lemma 2 one proves:

Theorem 7. *For the Boolean function g given by (1) the bound*

$$\overline{D}(g) \geq \frac{1}{\pi^2} n + o(n)$$

holds.

Finally, from Lemma 3 we see:

Theorem 8. *For the Boolean function g given by (1) the bounds*

$$\Delta(g) \geq \frac{1}{\pi^2} n + o(n) \qquad \text{and} \qquad \delta(g) \geq \frac{1}{6^{1/2}\pi} n^{1/2} + o(n^{1/2})$$

hold.

It is interesting to note that for representations of g as a polynomial over the field $GF(2)$ instead of over the reals the best known lower bound on the degree is of order $\Omega(\log n)$ (see [17]).

Finally, it is worth mentioning that since the average sensitivity of Boolean functions of the class $\mathbf{AC}^0$ does not exceed $(\log n)^{O(1)}$ (as it is shown in [14]), Theorem 5 provides an alternative proof for the statement proved in [3, 4] that g does not belong to $\mathbf{AC}^0$. This result has recently been improved in [1], where it is shown that for any prime p, testing square-free numbers as well as primality testing and testing co-primality of two given integers cannot be computed by $\mathbf{AC}^0[p]$ circuits, that is, $\mathbf{AC}^0$ circuits enhanced by $\textsc{Mod}_p$ gates.

Apparently the result of Lemma 4 can be improved by means of some more sophisticated sieve methods (see for instance [12]). However this will not improve our main results. On the other hand, we believe that one can prove the following asymptotic formula for the average sensitivity of testing square-free numbers.

Open Question 9. *Prove that*

$$s(g) = \frac{16}{\pi^2}\left(1 - \frac{8}{\pi^2}\right) n + o(n)$$

for the function g given by (1).

References

[1] E. Allender, M. Saks and I. E. Shparlinski, ‘A lower bound for primality’, *Proc. IEEE Conf. on Comp. Compl.*, IEEE, 1999 (to appear).

[2] A. Bernasconi, B. Codenotti and J. Simon, ‘On the Fourier analysis of Boolean functions’, *Preprint* (1996), 1-24.

[3] A. Bernasconi, C. Damm and I. E. Shparlinski, ‘Circuit and decision tree complexity of some number theoretic problems’, *Tech. Report 98-21*, Dept. of Math. and Comp. Sci., Univ. of Trier, 1998, 1–17.

[4] A. Bernasconi and I. E. Shparlinski, ‘Circuit complexity of testing square-free numbers’, *Proc. Intern. Symp. on Theor. Aspects of Comp. Sci.*, Springer-Verlag, Berlin, 1999 (to appear).

[5] R. B. Boppana, ‘The average sensitivity of bounded-depth circuits’, *Inform. Proc. Letters*, **63** (1997), 257–261.

[6] R. B. Boppana and M. Sipser, ‘The complexity of finite functions’, *Handbook of Theoretical Comp. Sci., Vol. A*, Elsevier, Amsterdam (1990), 757–804.

[7] Y. Brandman, A. Orlitsky and J. Hennessy, 'A spectral lower bound technique for the size of decision trees and two-level AND/OR circuits', *IEEE Transactions on Computers*, **39** (1990), 282–287.

[8] D. Coppersmith and I. E. Shparlinski, 'On polynomial approximation of the discrete logarithm and the Diffie–Hellman mapping', *J. Cryptology* (to appear).

[9] M. Dietzfelbinger, M. Kutyłowski and R. Reischuk, 'Feasible time-optimal algorithms for Boolean functions on exclusive-write parallel random access machine', *SIAM J. Comp.*, **25** (1996), 1196–1230.

[10] F. E. Fich, 'The complexity of computation on the parallel random access machine', *Handbook of Theoretical Comp. Sci., Vol. A*, Elsevier, Amsterdam (1990), 757–804.

[11] G. H. Hardy and E. M. Wright, *An introduction to the number theory*, Oxford Univ. Press, Oxford, 1965.

[12] D. R. Heath-Brown, 'The least square-free number in an arithmetic progression', *J. Reine Angew. Math.*, **332** (1982), 204–220.

[13] E. Koutsoupias, 'Improvements on Khrapchenko's theorem', *Theor. Comp. Sci.*, **116** (1993), 399–403.

[14] N. Linial, Y. Mansour and N. Nisan, 'Constant depth circuits, Fourier transform, and learnability', *Journal of the ACM*, **40** (1993), 607-620.

[15] N. Nisan and M. Szegedy, 'On the degree of Boolean functions as real polynomials', *Comp. Compl.*, **4** (1994), 301–313.

[16] I. Parberry and P. Yuan Yan, 'Improved upper and lower time bounds for parallel random access machines without simultaneous writes', *SIAM J. Comp.*, **20** (1991), 88–99.

[17] I. E. Shparlinski, 'On polynomial representations of Boolean functions related to some number theoretic problems', *Electronic Colloq. on Comp. Compl.*, `http://www.eccc.uni-trier.de/eccc/`, TR98-054, 1998, 1–13.

[18] I. E. Shparlinski, *Number theoretic methods in cryptography: Complexity lower bounds*, Birkhäuser, 1999.

[19] I. Wegener, *The complexity of Boolean functions*, Wiley-Teubner Series in Comp. Sci., Stuttgart (1987).