

# HOW LOW CAN APPROXIMATE DEGREE AND QUANTUM QUERY COMPLEXITY BE FOR TOTAL BOOLEAN FUNCTIONS?

ANDRIS AMBAINIS AND RONALD DE WOLF

March 27, 2014

**Abstract.** It has long been known that any Boolean function that depends on  $n$  input variables has both *degree* and *exact quantum query complexity* of  $\Omega(\log n)$ , and that this bound is achieved for some functions. In this paper we study the case of *approximate degree* and *bounded-error quantum query complexity*. We show that for these measures the correct lower bound is  $\Omega(\log n / \log \log n)$ , and we exhibit quantum algorithms for two functions where this bound is achieved.

**Keywords.** quantum computing, quantum algorithms, Boolean functions, polynomial approximations, computational complexity

**Subject classification.** 68Q12, 68Q17, 41A10, 06E30

## 1. Introduction

**1.1. Degree of Boolean functions.** The relations between Boolean functions and their representation as polynomials over various fields have long been studied and applied in areas like circuit complexity (Beigel (1993)), decision tree complexity (Buhrman & de Wolf (2002); Nisan & Szegedy (1994)) communication complexity (Buhrman & de Wolf (2001); Sherstov (2008)) and many others. In a seminal paper, Nisan & Szegedy (1994) made a systematic study of the representation and approximation of Boolean functions by real polynomials, focusing in particular on the *degree* of such polynomials. To state their and then our results, let us introduce some notation.

- Every function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  has a unique representation as an  $n$ -variate multilinear polynomial over the reals, i.e., there exist

real coefficients  $a_S$  such that  $f = \sum_{S \subseteq [n]} a_S \prod_{i \in S} x_i$ . Its *degree* is the number of variables in a largest monomial:  $\deg(f) := \max\{|S| : a_S \neq 0\}$ .

- We say  $g$   $\varepsilon$ -*approximates*  $f$  if  $|f(x) - g(x)| \leq \varepsilon$  for all  $x \in \{0, 1\}^n$ . The *approximate degree* of  $f$  is  $\deg(f) := \min\{\deg(g) : g \text{ } 1/3\text{-approximates } f\}$ . This is also sometimes called the “approximation degree”.
- For  $x \in \{0, 1\}^n$  and  $i \in [n]$ ,  $x^i$  is the input obtained from  $x$  by flipping the bit  $x_i$ . A variable  $x_i$  is called *sensitive* or *influential* on  $x$  (for  $f$ ) if  $f(x) \neq f(x^i)$ . In this case we also say  $f$  *depends* on  $x_i$ . The *influence* of  $x_i$  (on Boolean function  $f$ ) is the fraction of inputs  $x \in \{0, 1\}^n$  where  $i$  is influential:  $\text{Inf}_i(f) := \Pr_x[f(x) \neq f(x^i)]$ , where the subscript on the right-hand side denotes probability taken over uniformly distributed  $x \in \{0, 1\}^n$ .
- The *sensitivity*  $s(f, x)$  of  $f$  at input  $x$  is the number of variables that are influential on  $x$ , and the sensitivity of  $f$  is  $s(f) := \max_{x \in \{0, 1\}^n} s(f, x)$ .

One of the main results of Nisan & Szegedy (1994) is that every function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  that depends on all  $n$  variables has degree  $\deg(f) \geq \log n - O(\log \log n)$  (our logarithms are to base 2). Their proof goes as follows. On the one hand, the function  $f_i(x) := f(x) - f(x^i)$  is a polynomial of degree at most  $\deg(f)$  that is not identically equal to 0. Hence by a version of the Schwartz-Zippel lemma,  $f_i$  is nonzero on at least a  $2^{-\deg(f)}$ -fraction of the Boolean cube. Since  $f_i(x) \neq 0$  iff  $i$  is sensitive on  $x$ , this shows

$$(1.1) \quad \text{Inf}_i(f) \geq 2^{-\deg(f)} \text{ for every influential } x_i.$$

On the other hand, with a bit of Fourier analysis (see Section 2.1) one can show

$$\sum_{i=1}^n \text{Inf}_i(f) \leq \deg(f)$$

and hence

$$(1.2) \quad \text{there is an influential } x_i \text{ with } \text{Inf}_i(f) \leq \deg(f)/n.$$

Combining (1.1) and (1.2) implies  $\deg(f) \geq \log n - O(\log \log n)$ . As Nisan and Szegedy observe, this lower bound is tight up to the  $O(\log \log n)$  term for the *Address function* (which was already used by Simon (1983)): let  $k$  be some power of 2,  $n = k + \log k$ , and view the last  $\log k$  bits of the  $n$ -bit input as an address in the first  $k$  bits. Define  $f(x)$  as the value of the addressed variable. This function depends on all  $n$  variables and has degree  $\log k + 1 \leq \log n + 1$ , because we can write it as a sum over all  $\log k$ -bit addresses, multiplied by the addressed variable. More explicitly: if we write the input as  $x = zy$  with  $z \in \{0, 1\}^k$  and  $y \in \{0, 1\}^{\log k}$ , the polynomial representation of the Address function is

$$\sum_{a \in \{0, 1\}^{\log k}} z_a \prod_{i: a_i=1} y_i \prod_{i: a_i=0} (1 - y_i).$$

**1.2. Approximate degree of Boolean functions.** Our focus in this paper is on what happens if instead of considering *representation* by polynomials we consider *approximation* by polynomials. While Nisan and Szegedy studied some properties of approximate degree in their paper, they did not state a general lower bound for all functions depending on  $n$  variables. Can we modify their proof to work for approximating polynomials? While (1.2) still holds if we replace the right-hand side by approximate degree, (1.1) becomes much weaker. Since it is known that  $\text{Inf}_i(f) \geq 2^{-2s(f)+1}$  (Simon 1983, p. 443) and  $s(f) = O(\widetilde{\deg}(f)^2)$  (Nisan & Szegedy (1994)), we have

$$(1.3) \quad \text{Inf}_i(f) \geq 2^{-O(\widetilde{\deg}(f)^2)} \text{ for every influential } x_i.$$

This lower bound on  $\text{Inf}_i(f)$  is in fact optimal. For example for the  $n$ -bit OR-function each variable has influence  $(n+1)/2^n$  and the approximate degree is  $\Theta(\sqrt{n})$ . Hence adapting Nisan and Szegedy's proof of the  $\Omega(\log n)$  lower bound on exact degree will only give an  $\Omega(\sqrt{\log n})$  bound on approximate degree. Another way to prove that same bound is to use the facts that  $s(f) = O(\widetilde{\deg}(f)^2)$  and  $s(f) = \Omega(\log n)$  if  $f$  depends on  $n$  bits (Simon (1983)).

In Section 2 we improve this bound to  $\Omega(\log n / \log \log n)$ . The proof idea is the following. Suppose  $P$  is a degree- $d$  polynomial that approximates  $f$ . First, by a bit of Fourier analysis we show that there

is a variable  $x_i$  such that the function  $P_i(x) := P(x) - P(x^i)$  (which has degree  $\leq d$  and expectation 0) has low variance. We then use a concentration result for low-degree polynomials to show that  $P_i$  is close to its expectation for almost all of the inputs. On the other hand, since  $x_i$  has nonzero influence, (1.3) implies that  $|P_i|$  must be close to 1 (and hence far from its expectation) on at least a  $2^{-O(d^2)}$ -fraction of all inputs. Combining these things then yields  $d = \Omega(\log n / \log \log n)$ .

**1.3. Relation with quantum query complexity.** One of the main reasons that the degree and approximate degree of a Boolean function are interesting measures, is their relation to the *quantum query complexity* of that function. We define  $Q_E(f)$  and  $Q_2(f)$  as the minimal query complexity of *exact* (errorless) and  $1/3$ -error quantum algorithms for computing  $f$ , respectively, referring to Buhrman & de Wolf (2002) for precise definitions.

Beals *et al.* (2001) established the following lower bounds on quantum query complexity in terms of degrees:

$$Q_E(f) \geq \deg(f)/2 \quad \text{and} \quad Q_2(f) \geq \widetilde{\deg}(f)/2.$$

They also proved that classical deterministic query complexity is at most  $O(\widetilde{\deg}(f)^6)$ , improving an earlier 8th-power result of Nisan & Szegedy (1994), so this lower bound is never more than a polynomial off for total Boolean functions. While the polynomial method sometimes gives bounds that are polynomially weaker than the true complexity (Ambainis (2006)), still many tight quantum lower bounds are based on this method (Aaronson & Shi (2004); Klauck *et al.* (2007)).

The *classical* bounded-error query complexity is lower bounded by sensitivity (Nisan & Szegedy (1994)) and hence always  $\Omega(\log n)$ . In contrast, our new lower bound on approximate degree implies that  $Q_2(f) = \Omega(\log n / \log \log n)$  for all total Boolean functions that depend on  $n$  variables. In Section 3 we construct two functions that meet this bound, showing that  $Q_2(f)$  can be  $O(\log n / \log \log n)$  for a total function that depends on  $n$  bits. Since  $Q_2(f) \geq \widetilde{\deg}(f)/2$ , we immediately also get that  $\widetilde{\deg}(f)$  can be  $O(\log n / \log \log n)$ . Interestingly, the only way we know to construct  $f$  with asymptotically minimal  $\widetilde{\deg}(f)$  is through such quantum algorithms—this fits into the growing sequence of classical results proven by quantum means (Drucker &

de Wolf (2011)).

The idea behind our construction is to modify the Address function (which achieves the smallest degree in the exact case). Let  $n = k + m$ . We use the last  $m$  bits of the input to build a *quantum addressing scheme* that specifies an address in the first  $k$  bits. The value of the function is then defined to be the value of the addressed bit. The following requirements need to be met by the addressing scheme:

- There is a quantum algorithm to compute the index  $i$  addressed by  $y \in \{0, 1\}^m$ , using few queries to  $y$ ;
- For every index  $i \in \{1, \dots, k\}$ , there is a string  $y \in \{0, 1\}^m$  that addresses  $i$  (so that the function depends on all of the first  $k$  bits);
- Every string  $y \in \{0, 1\}^m$  addresses one of  $1, \dots, k$  (so the resulting function on  $k + m$  bits is total);

In Section 3 we give two constructions of addressing schemes that address  $k = d^{\Theta(d)}$  bits using  $d$  quantum queries. Each construction gives a total Boolean function on  $n \geq d^{\Theta(d)}$  bits that is computable with  $d + 1 = O(\log n / \log \log n)$  quantum queries:  $d$  queries for computing the address  $i$  and 1 query to retrieve the addressed bit  $x_i$ .<sup>1</sup>

To summarize, all total Boolean functions that depend on  $n$  variables have approximate degree and bounded-error quantum query complexity at least  $\Omega(\log n / \log \log n)$ , and that lower bound is tight for some functions.

## 2. Approximate degree is $\Omega(\log n / \log \log n)$ for all total $f$

**2.1. Tools from Fourier analysis.** We use the framework of Fourier analysis on the Boolean cube. We will just introduce what we need

---

<sup>1</sup>It is interesting to contrast this with “quantum oracle interrogation” (van Dam (1998)). An arbitrary  $m$ -bit string can be recovered using roughly  $m/2$  quantum queries (van Dam (1998)), but not less (Ambainis *et al.* (2013)). In other words,  $d$  quantum queries can recover an address of roughly  $2d$  bits. In the addressing schemes we consider here, where different  $m$ -bit strings can point to the same address,  $d$  quantum queries can recover an address of roughly  $d \log d$  bits encoded in a larger  $m$ -bit string.

here, referring to O’Donnell (2008, 2014); de Wolf (2008) for more details and references. In this section it will be convenient to denote bits as  $+1$  and  $-1$ , so a Boolean function will now be  $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ . Unless mentioned otherwise, expectations and probabilities below are taken over a uniformly random  $x \in \{\pm 1\}^n$ .

Define the inner product between functions  $f, g : \{\pm 1\}^n \rightarrow \mathbb{R}$  as

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} f(x)g(x) = \mathbb{E}[f \cdot g].$$

For  $S \subseteq [n]$ , the function  $\chi_S$  is the product (parity) of the variables indexed in  $S$ . These functions form an orthonormal basis for the space of all real-valued functions on the Boolean cube. The *Fourier coefficients* of  $f$  are  $\widehat{f}(S) = \langle f, \chi_S \rangle$ , and we can write  $f$  in its Fourier decomposition

$$f = \sum_{S \subseteq [n]} \widehat{f}(S) \chi_S.$$

The *degree*  $\deg(f)$  of  $f$  is  $\max\{|S| : \widehat{f}(S) \neq 0\}$ . The *expectation* or *average* of  $f$  is  $\mathbb{E}[f] = \widehat{f}(\emptyset)$ , and its *variance* is  $\text{Var}[f] = \mathbb{E}[f^2] - \mathbb{E}[f]^2 = \sum_{S \neq \emptyset} \widehat{f}(S)^2$ . For  $p \geq 1$ , the *p-norm* of  $f$  is defined as

$$\|f\|_p = \mathbb{E}[|f|^p]^{1/p}.$$

This is monotone non-decreasing in  $p$ . For  $p = 2$ , Parseval’s identity says

$$\|f\|_2^2 = \sum_S \widehat{f}(S)^2.$$

For low-degree  $f$ , the famous *hypercontractive inequality* (sometimes called the *Bonami-Beckner inequality*) implies that higher norms cannot be *much* bigger than the 2-norm.<sup>2</sup>

**THEOREM 2.1.** *Let  $f$  be a multilinear  $n$ -variate polynomial. If  $q \geq 2$ , then*

$$\|f\|_q \leq (q - 1)^{\deg(f)/2} \|f\|_2.$$

---

<sup>2</sup>See for example (O’Donnell 2014, Section 9.5) or (de Wolf 2008, after Theorem 4.1) for a derivation of this statement from the hypercontractive inequality. See (O’Donnell 2014, Chapters 9 and 10) or (Janson 1997, Chapter 5) for more background on hypercontractivity.

The main tool we use is the following concentration result for degree- $d$  polynomials (the degree-1 case is essentially the familiar Chernoff bound). Its derivation from Theorem 2.1 is folklore, see for example (Dinur *et al.* 2007, Section 2.2) or (O’Donnell 2008, Theorem 5.4). For completeness we include the proof below.

**THEOREM 2.2.** *Let  $F$  be a multilinear  $n$ -variate polynomial of degree at most  $d$ , with expectation 0 and variance  $\sigma^2 = \|F\|_2^2$ . For all  $t \geq (2e)^{d/2}$  it holds that*

$$\Pr[|F| \geq t\sigma] \leq \exp\left(-\frac{d}{2e} \cdot t^{2/d}\right).$$

**PROOF.** Theorem 2.1 implies

$$\mathbb{E}[|F|^q] = \|F\|_q^q \leq (q-1)^{dq/2} \|F\|_2^q = (q-1)^{dq/2} \sigma^q.$$

Using Markov’s inequality gives

$$\Pr[|F| \geq t\sigma] = \Pr[|F|^q \geq (t\sigma)^q] \leq \frac{\mathbb{E}[|F|^q]}{(t\sigma)^q} \leq \frac{(q-1)^{dq/2} \sigma^q}{(t\sigma)^q} \leq \frac{q^{dq/2}}{t^q}.$$

Choosing  $q = t^{2/d}/e$  gives the theorem (note that our assumption on  $t$  implies  $q \geq 2$ ).  $\square$

**2.2. The lower bound proof.** Here we prove our main lower bound.

**THEOREM 2.3.** *Every Boolean function  $f$  that depends on  $n$  input bits has*

$$\widetilde{\deg}(f) = \Omega(\log n / \log \log n).$$

**PROOF.** Let  $P : \mathbb{R}^n \rightarrow [-1, 1]$  be a  $1/3$ -approximating polynomial for  $f$ . We assume that the range of  $f$  is  $[-1, 1]$  rather than  $[-4/3, 4/3]$ . This does not change anything significant in the result but simplifies the calculations (by avoiding some factors of  $4/3$  in various places below). Our goal is to show that  $d := \deg(P)$  is  $\Omega(\log n / \log \log n)$ . If  $d > \log n / \log \log n$  then we are already done, so assume  $d \leq \log n / \log \log n$ .

For each  $i \in [n]$ , we define  $f_i$  by  $f_i(x) = (f(x) - f(x^i))/2$ , and  $P_i$  by  $P_i(x) = (P(x) - P(x^i))/2$ . Note that both  $f_i$  and  $P_i$  have expectation 0. We have  $f_i(x) \in \{\pm 1\}$  if  $i$  is sensitive for  $x$ , and  $f_i(x) = 0$  otherwise. Similarly we have  $P_i(x) \in [-1, -2/3] \cup [2/3, 1]$  if  $i$  is sensitive

for  $x$ , and  $P_i(x) \in [-1/3, 1/3]$  otherwise. The Fourier decomposition of  $P_i$  is

$$\begin{aligned} P_i(x) &= \frac{1}{2} \left( \sum_S \widehat{P}(S) \chi_S(x) - \sum_S \widehat{P}(S) \chi_S(x^i) \right) \\ &= \frac{1}{2} \left( \sum_S \widehat{P}(S) \chi_S(x) - \sum_S \widehat{P}(S) (-1)^{|i \in S|} \chi_S(x) \right) \\ &= \frac{1}{2} \sum_S (1 - (-1)^{|i \in S|}) \widehat{P}(S) \chi_S(x). \end{aligned}$$

Hence  $\widehat{P}_i(S) = \widehat{P}(S)$  if  $i \in S$  and  $\widehat{P}_i(S) = 0$  if  $i \notin S$ . Then

$$\begin{aligned} \sum_{i=1}^n \|P_i\|_2^2 &= \sum_{i=1}^n \sum_S \widehat{P}_i(S)^2 = \sum_{i=1}^n \sum_{S \ni i} \widehat{P}(S)^2 = \sum_S |S| \widehat{P}(S)^2 \\ &\leq d \sum_S \widehat{P}(S)^2 = d \|P\|_2^2 \leq d. \end{aligned}$$

Therefore there exists an  $i \in [n]$  for which

$$\|P_i\|_2^2 \leq d/n.$$

Assume that  $i = 1$  is one such index. Because every variable (including  $x_1$ ) is influential, Eq. (1.3) implies

$$\text{Inf}_1(f) \geq 2^{-O(d^2)}.$$

Define  $\sigma^2 = \text{Var}[P_1]$ , and note that this variance is not very large:  $\sigma^2 = \|P_1\|_2^2 \leq d/n$ . Set  $t = 1/2\sigma \geq \sqrt{n/4d}$ . Then  $t \geq (2e)^{d/2}$  for sufficiently large  $n$ , because we assumed  $d \leq \log n / \log \log n$ . Now use Theorem 2.2 to get

$$\begin{aligned} \text{Inf}_1(f) &= \Pr[f_1(x) \in \{\pm 1\}] \\ &= \Pr[|P_1(x)| \geq 1/2] \\ &= \Pr[|P_1(x)| \geq t\sigma] \\ &\leq \exp\left(-\frac{d}{2e} \cdot t^{2/d}\right) \\ &\leq \exp\left(-\frac{d}{2e} \cdot (n/4d)^{1/d}\right). \end{aligned}$$



Combining the upper and lower bounds on  $\text{Inf}_1(f)$  gives

$$2^{-O(d^2)} \leq \exp\left(-(d/2e)(n/4d)^{1/d}\right).$$

Taking logarithms of left and right-hand side and negating gives

$$O(d^2) \geq (d/2e)(n/4d)^{1/d}.$$

Dividing by  $d$  and using our assumption that  $d \leq \log n / \log \log n$  implies, for sufficiently large  $n$ :

$$\log n \geq (n/4d)^{1/d}.$$

Taking logarithms once more we get

$$d \geq \log(n/4d) / \log \log n = \log n / \log \log n - O(1),$$

which proves the theorem. □

Note that the constant factor in the  $\Omega(\cdot)$  is essentially 1 for any constant approximation error. The  $\Omega(\log n / \log \log n)$  bound remains valid even for quite large errors: the same proof shows that for every constant  $\gamma < 1/2$ , every polynomial  $P$  for which  $\text{sgn}(P(x)) = f(x)$  and  $|P(x)| \in [1/n^\gamma, 1]$  for all  $x \in \{\pm 1\}^n$ , has degree  $\Omega(\log n / \log \log n)$ . This lower bound no longer holds if  $\gamma = 1$ ; for example for odd  $n$ , the degree-1 polynomial  $\sum_{i=1}^n x_i/n$  has the same sign as the majority function, and  $|P(x)| \in [1/n, 1]$  everywhere.

### 3. Functions with quantum query complexity

$$O(\log n / \log \log n)$$

In this section we exhibit two  $n$ -bit Boolean functions whose bounded-error quantum query complexity (and hence approximate degree) is  $O(\log n / \log \log n)$ .

**THEOREM 3.1.** *There is a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  that depends on all  $n$  variables and has*

$$Q_2(f) = O\left(\frac{\log n}{\log \log n}\right).$$

**PROOF.** Let us call a function  $a(x_1, \dots, x_m)$  of  $m$  variables  $x_1, \dots, x_m \in \{0, 1\}$  a  $k$ -addressing scheme if  $a(x_1, \dots, x_m) \in [k]$  and, for every  $i \in [k]$ , there exist  $x_1, \dots, x_m \in \{0, 1\}$  such that  $a(x_1, \dots, x_m) = i$ .

LEMMA 3.2. *For every  $t > 0$ , with  $k = t^t$  and  $m = t^2$ , there exists a  $k$ -addressing scheme  $a(x_1, \dots, x_m)$  that can be computed with error probability  $\leq 1/3$  using  $O(t)$  quantum queries.*

PROOF. In Section 3.1 and Section 3.2 we give two constructions of addressing schemes achieving this bound.  $\square$

First assume for simplicity that  $n$  is of the form  $n = t^t + t^2$  for some integer  $t$ . Set  $k = t^t$  and  $m = t^2$ . Without loss of generality, we assume the  $k$ -addressing scheme  $a(x_1, \dots, x_m)$  from Lemma 3.2 depends on all variables  $x_1, \dots, x_m$ . The last  $m$  input bits will provide an address of one of the first  $k$  input bits. Define the following  $n$ -bit Boolean function:

$$f(x_1, \dots, x_n) = x_{a(x_{k+1}, x_{k+2}, \dots, x_{k+m})}.$$

Since  $a(x_1, \dots, x_m)$  can be computed with  $O(t)$  quantum queries, we have that  $f(x_1, \dots, x_n)$  can be computed with  $O(t) + 1$  queries. Its number of variables is  $n > k = t^t$ . Hence,

$$\frac{\log n}{\log \log n} \geq \frac{t \log t}{\log t + \log \log t} = (1 + o(1))t.$$

If  $n$  is not of the form  $n = t^t + t^2$ , then let  $t$  be the smallest integer such that  $n < t^t + t^2$ . The above construction gives an  $m$ -bit  $k$ -addressing scheme  $a$  (with  $k = t^t$  and  $m = t^2$ ), and a Boolean function  $f$  on  $N := k + m$  bits. We will now reduce the number of input bits from  $N$  to  $n$  by dropping the last  $N - n$  input bits of the  $k$ -bit part. More precisely, define a new addressing scheme  $a'$  such that  $a'(x_1, \dots, x_m) = 1$  if  $a(x_1, \dots, x_m) > N - n$ , and  $a' = a$  otherwise. This new scheme  $a'$  now addresses  $k' := k - (N - n)$  bits, and without loss of generality  $a'$  still depends on all  $m$  input bits. The induced Boolean function  $f'(x_1, \dots, x_n) = x_{a'(x_{k'+1}, x_{k'+2}, \dots, x_{k'+m})}$  depends on  $k' + m = k - (N - n) + m = n$  input bits and is computable using  $O(t) + 1 = O(\log n / \log \log n)$  quantum queries.  $\square$

**3.1. Addressing scheme: 1st construction.** Our first addressing scheme is based on the Bernstein-Vazirani algorithm (Bernstein & Vazirani (1997)). For simplicity assume  $t$  is a power of 2. For a string  $z \in \{0, 1\}^{\log t}$ , let  $h(z)$  be its  $t$ -bit Hadamard codeword:  $h(z)_j = z \cdot j$

mod 2, where  $j$  ranges over  $\{0, 1\}^{\log t}$ , and  $z \cdot j$  denotes inner product. The Bernstein-Vazirani algorithm recovers  $z$  with probability 1 using only one quantum query if its  $t$ -bit input is of the form  $h(z)$ . The  $m$ -bit input  $x$  to the addressing scheme consists of  $t$  blocks  $x^{(1)}, \dots, x^{(t)}$  of  $t$  bits each, so  $m = t^2$ . Define the addressing scheme as follows:

If  $x$  is of the form  $h(z^{(1)}) \dots h(z^{(t)})$  then set  $a(x) := z^{(1)} \dots z^{(t)}$ .  
 Otherwise set  $a(x) := 0^{t \log t}$ .

Note that the value of  $a(x)$  is a  $t \log t$ -bit string, and that the function is surjective. Hence the function  $a$  addresses a space of  $k = 2^{t \log t} = t^t$  bits.

The following algorithm computes  $a(x)$  with  $O(t)$  quantum queries:

1. Use the Bernstein-Vazirani algorithm  $t$  times, once on each  $x^{(j)}$ , with outputs  $z^{(1)}, \dots, z^{(t)} \in \{0, 1\}^{\log t}$ .
2. Use Grover's search algorithm (Brassard *et al.* (2002); Grover (1996)) to check if  $x = x^{(1)} \dots x^{(t)}$  equals  $h(z^{(1)}) \dots h(z^{(t)})$ , i.e., to search for a bit-position where these two  $m$ -bit strings are different.
3. If yes, output  $a(x) = z^{(1)} \dots z^{(t)}$ . Else output  $0^{t \log t}$ .

The query complexity is  $t$  queries for the first step and  $O(\sqrt{m}) = O(t)$  for the second.

If the input  $x$  is the concatenation of  $t$  Hadamard codewords  $h(z^{(1)}), \dots, h(z^{(t)})$ , then the first step will identify the correct  $z^{(1)}, \dots, z^{(t)}$  with probability 1, and the second step will not find any discrepancy.<sup>3</sup> On the other hand, if the input is not the concatenation of  $t$  Hadamard codewords then the two strings compared in step 2 are not equal, and Grover search will find a discrepancy with probability at least  $2/3$ , in which case the algorithm outputs the correct value  $0^{t \log t}$ .

---

<sup>3</sup>Note that the probability that a random  $x \in \{0, 1\}^{t^2}$  is of this concatenated-Hadamard form is  $t^t / 2^{t^2} = 2^{-(1-o(1))t^2}$ . Since the Boolean function  $f$  based on this addressing scheme will have approximate degree  $O(t)$ , the influence of each of the  $t^2$  address bits on  $f$  matches the lower bound of Eq. (1.3). Whenever the address bits are not of the concatenated-Hadamard form, the address  $0^{t \log t}$  points to the first bit of the input, so this variable has influence nearly 1 on  $f$ . The probability that any of the 2nd to  $k$ th bits are addressed is exactly  $2^{-t^2}$ , so their influences are  $2^{-t^2}$ , again matching Eq. (1.3).

**3.2. Addressing scheme: 2nd construction.** The second addressing scheme is defined as follows. We select  $k = t^t$  words  $w^{(i)}$  of  $m = t^2$  bits each, such that any two distinct words  $w^{(i)}$  and  $w^{(j)}$  have Hamming distance in the interval  $I = [\frac{m}{2} - ct\sqrt{t \log t}, \frac{m}{2} + ct\sqrt{t \log t}]$ .

One can for example show the existence of such strings using a standard application of the probabilistic method, as follows. Select the  $w^{(i)}$  randomly from  $\{0, 1\}^m$ . For distinct  $i$  and  $j$ , the expected Hamming distance between  $w^{(i)}$  and  $w^{(j)}$  equals  $m/2$ . By a Chernoff bound, the probability that this Hamming distance is outside of the interval  $I$  is  $2^{-\Omega(c^2 t^3 \log(t)/m)} = 2^{-\Omega(c^2 t \log t)}$ . If we choose  $c$  a sufficiently large constant then this probability is  $o(1/\binom{k}{2})$ . Since there are  $\binom{k}{2}$  distinct  $i, j$ -pairs, the union bound implies that with probability  $1 - o(1)$ , all pairs of words  $w^{(i)}$  and  $w^{(j)}$  have Hamming distance in the interval  $I$ .

For input  $x \in \{0, 1\}^m$ , define  $a(x) := i$  if  $x = w^{(i)}$ , and  $a(x) := 1$  if  $x$  does not equal any of  $w^{(1)}, \dots, w^{(k)}$ . We select  $t' = O(t)$  so that

$$\left(\frac{2c\sqrt{\log t}}{\sqrt{t}}\right)^{t'} \leq \frac{1}{t^{2t}}.$$

Let

$$|\psi\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^m (-1)^{x_j} |j\rangle.$$

Let  $|\psi_i\rangle$  be the state  $|\psi\rangle$  defined above if  $x = w^{(i)}$ . If  $i \neq j$ , we have

$$\langle \psi_i^{\otimes t'} | \psi_j^{\otimes t'} \rangle = (\langle \psi_i | \psi_j \rangle)^{t'} \leq \left(\frac{2c\sqrt{\log t}}{\sqrt{t}}\right)^{t'} \leq \frac{1}{t^{2t}}.$$

The following lemma is quantum computing folklore. For the sake of completeness we give a proof.

**LEMMA 3.3.** *Let  $k \geq 1$  and  $|\phi_1\rangle, \dots, |\phi_k\rangle$  be states such that  $|\langle \phi_i | \phi_j \rangle| \leq 1/k^2$  whenever  $i \neq j$ . Then there is a measurement that, given  $|\phi_i\rangle$ , produces outcome  $i$  with probability at least  $2/3$ .*

**PROOF.** The lemma is obvious for  $k = 1$ , so we can assume  $k \geq 2$ . Let Hilbert space  $\mathcal{H}$  be the span of the states  $|\phi_1\rangle, \dots, |\phi_k\rangle$ , and define  $A = \sum_{i=1}^k |\phi_i\rangle\langle \phi_i|$  as an operator on this space. We want to show that

$A$  is close to the identity operator on  $\mathcal{H}$ . We first show that  $A|\phi_j\rangle$  is close to  $|\phi_j\rangle$  for all  $j \in [k]$ . Define  $|\delta_j\rangle = A|\phi_j\rangle - |\phi_j\rangle$ . We have

$$\| |\delta_j\rangle \| = \left\| \sum_{i \in [k] \setminus \{j\}} |\phi_i\rangle \langle \phi_i | \phi_j \rangle \right\| \leq \sum_{i \in [k] \setminus \{j\}} |\langle \phi_i | \phi_j \rangle| \leq \frac{k-1}{k^2}.$$

Now we show  $A|v\rangle$  is close to  $|v\rangle$  for an arbitrary unit vector  $|v\rangle = \sum_{j=1}^k \alpha_j |\phi_j\rangle$  in  $\mathcal{H}$ . Define  $a := \sum_{j=1}^k |\alpha_j|^2$ . We have

$$1 = \langle v | v \rangle = \sum_{i,j=1}^k \alpha_i^* \alpha_j \langle \phi_i | \phi_j \rangle = a + \sum_{i \neq j} \alpha_i^* \alpha_j \langle \phi_i | \phi_j \rangle.$$

Also, using the Cauchy-Schwarz inequality,

$$\begin{aligned} \sum_{i \neq j} \alpha_i^* \alpha_j \langle \phi_i | \phi_j \rangle &\leq \sqrt{\sum_{i \neq j} |\alpha_i|^2 |\alpha_j|^2} \sqrt{\sum_{i \neq j} |\langle \phi_i | \phi_j \rangle|^2} \\ &\leq \sqrt{\sum_{i,j} |\alpha_i|^2 |\alpha_j|^2} \sqrt{\sum_{i,j} 1/k^4} = a/k. \end{aligned}$$

This implies  $1 \geq a - a/k$  and hence  $a \leq 1/(1 - 1/k) = k/(k-1)$ . We have

$$A|v\rangle = \sum_{j=1}^k \alpha_j A|\phi_j\rangle = \sum_{j=1}^k \alpha_j (|\phi_j\rangle + |\delta_j\rangle) = |v\rangle + \sum_{j=1}^k \alpha_j |\delta_j\rangle.$$

This implies, again using Cauchy-Schwarz,

$$\begin{aligned} \|A|v\rangle - |v\rangle\| &\leq \sum_{j=1}^k |\alpha_j| \| |\delta_j\rangle \| \leq \sqrt{\sum_{j=1}^k |\alpha_j|^2} \sqrt{\sum_{j=1}^k \| |\delta_j\rangle \|^2} \\ &\leq \sqrt{\frac{k}{k-1}} \sqrt{\frac{k(k-1)^2}{k^4}} = \sqrt{\frac{k-1}{k^2}} \leq \frac{1}{2}. \end{aligned}$$

Hence  $A \leq \frac{3}{2}I$ .

Our measurement will consist of the operators  $E_i = \frac{2}{3}|\phi_i\rangle\langle\phi_i|$  for all  $i \in [k]$ , and  $E_0 = I - \sum_{i=1}^k E_i$ . By the previous discussion  $E_0 = I - \frac{2}{3}A \geq 0$ , so  $\{E_i\}_{i=0}^k$  is a well-defined measurement (more precisely, a POVM). Given state  $|\phi_i\rangle$ ,  $i \in [k]$ , the probability that our measurement produces the correct outcome  $i$  equals  $\text{Tr}(E_i|\phi_i\rangle\langle\phi_i|) = 2/3$ .  $\square$

We will apply this lemma to the  $k$  states  $|\phi_i\rangle = |\psi_i\rangle^{\otimes t'}$ . Our  $O(t)$ -query quantum algorithm is as follows:

1. Use  $t' = O(t)$  queries to generate  $|\psi\rangle^{\otimes t'}$ .
2. Apply the measurement of Lemma 3.3.
3. If the measurement gives some  $i \neq 1$ , then use Grover's search algorithm (Brassard *et al.* (2002); Grover (1996)) (with error probability  $\leq 1/3$ ) to search for  $j \in [m]$  such that  $x_j \neq w_j^{(i)}$ .
4. If no such  $j$  is found, then output  $i$ . Else output 1.

The number of queries is  $O(t)$  to generate  $|\psi\rangle^{\otimes t'}$  and  $O(\sqrt{m}) = O(t)$  for Grover search, so  $O(t)$  in total.

If the input  $x$  equals some  $w^{(i)}$ , then the measurement of Lemma 3.3 will produce the correct  $i$  with probability at least  $2/3$  and Grover search will not find  $j$  s.t.  $x_j \neq w_j^{(i)}$ . Hence, the whole algorithm will output  $i$  with probability at least  $2/3$ . If the input  $x$  is not equal to any  $w^{(i)}$ , then the measurement will produce some  $i$  but Grover search will find  $j$  s.t.  $x_j \neq w_j^{(i)}$ , with probability at least  $2/3$ . As a result, the algorithm will output the correct answer 1 with probability at least  $2/3$  in this case.

## 4. Conclusion

We analyzed how low approximate degree and bounded-error quantum query complexity can be for total Boolean functions that depend on  $n$  bits, and gave an optimal answer to this question. We proved a general lower bound of  $\Omega(\log n / \log \log n)$  on approximate degree (and hence also on quantum query complexity), by combining a tight lower bound on influence in terms of approximate degree (Eq. (1.3)) with the hypercontractive inequality. We also exhibited functions where this bound is achieved. These functions are variations of the Address function that have lower quantum query complexity, based on addressing schemes where  $d \log d$  address-bits can be recovered using  $O(d)$  quantum queries.

## Acknowledgements

Eq. (1.3) was observed in email discussion between RdW and Scott Aaronson in 2008. We thank Artūrs Bačkurs, Oded Regev, Mario Szegedy, and the anonymous referees of CCC'13 and Computational Complexity for useful comments and references.

This research was supported by the European Commission under the projects QCS (Grant No. 255961) and QALGO (Grant No. 600700). Andris Ambainis was also supported by the ERC Advanced Grant MQC. Ronald de Wolf was supported by a Vidi grant from the Netherlands Organization for Scientific Research (NWO), and by the ERC Consolidator Grant QPROGRESS.

## References

- S. AARONSON & Y. SHI (2004). Quantum Lower Bounds for the Collision and the Element Distinctness Problems. *Journal of the ACM* **51**(4), 595–605.
- A. AMBAINIS (2006). Polynomial degree vs. quantum query complexity. *Journal of Computer and System Sciences* **72**(2), 220–238. Earlier version in FOCS'03. quant-ph/0305028.
- A. AMBAINIS, A. BAČKURS, J. SMOTROVS & R. DE WOLF (2013). Optimal quantum query bounds for almost all Boolean functions. In *Proceedings of 30th Annual Symposium on Theoretical Aspects of Computer Science (STACS'13)*, 446–453. Quant-ph/1208.1122.
- R. BEALS, H. BUHRMAN, R. CLEVE, M. MOSCA & R. DE WOLF (2001). Quantum Lower Bounds by Polynomials. *Journal of the ACM* **48**(4), 778–797. Earlier version in FOCS'98. quant-ph/9802049.
- R. BEIGEL (1993). The Polynomial Method in Circuit Complexity. In *Proceedings of the 8th IEEE Structure in Complexity Theory Conference*, 82–95.
- E. BERNSTEIN & U. VAZIRANI (1997). Quantum Complexity Theory. *SIAM Journal on Computing* **26**(5), 1411–1473. Earlier version in STOC'93.
- G. BRASSARD, P. HØYER, M. MOSCA & A. TAPP (2002). Quantum Amplitude Amplification and Estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, 53–74. Quant-ph/0005055.

- H. BUHRMAN & R. DE WOLF (2001). Communication Complexity Lower Bounds by Polynomials. In *Proceedings of 16th IEEE Conference on Computational Complexity*, 120–130. Cs.CC/9910010.
- H. BUHRMAN & R. DE WOLF (2002). Complexity Measures and Decision Tree Complexity: A Survey. *Theoretical Computer Science* **288**(1), 21–43.
- W. VAN DAM (1998). Quantum Oracle Interrogation: Getting all information for almost half the price. In *Proceedings of 39th IEEE FOCS*, 362–367. Quant-ph/9805006.
- I. DINUR, E. FRIEDGUT, G. KINDLER & R. O’DONNELL (2007). On the Fourier tails of bounded functions over the discrete cube. *Israel Journal of Mathematics* **160**(1), 389–412. Earlier version in STOC’06.
- A. DRUCKER & R. DE WOLF (2011). Quantum proofs for classical theorems. *Theory of Computing ToC Library, Graduate Surveys 2*.
- L. K. GROVER (1996). A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of 28th ACM STOC*, 212–219. Quant-ph/9605043.
- S. JANSON (1997). *Gaussian Hilbert Spaces*, volume 129 of *Cambridge Tracts in Mathematics*. Cambridge University Press.
- H. KLAUCK, R. ŠPALEK & R. DE WOLF (2007). Quantum and Classical Strong Direct Product Theorems and Optimal Time-Space Tradeoffs. *SIAM Journal on Computing* **36**(5), 1472–1493. Earlier version in FOCS’04. quant-ph/0402123.
- N. NISAN & M. SZEGEDY (1994). On the Degree of Boolean functions as Real Polynomials. *Computational Complexity* **4**(4), 301–313. Earlier version in STOC’92.
- R. O’DONNELL (2008). Some Topics in Analysis of Boolean Functions. Technical report, ECC Report TR08–055. Paper for an invited talk at STOC’08.
- R. O’DONNELL (2014). *Analysis of Boolean functions*. Cambridge University Press. Preliminary version available at <http://analysisofbooleanfunctions.org/>.
- A. SHERSTOV (2008). Communication Lower Bounds Using Dual Polynomials. *Bulletin of the EATCS* **95**, 59–93.



H. U. SIMON (1983). A Tight  $\Omega(\log \log n)$ -bound on the Time for Parallel RAM's to Compute Non-Degenerate Boolean Functions. In *Symposium on Foundations of Computation Theory*, volume 158 of *Lecture Notes in Computer Science*, 439–444. Springer.

R. DE WOLF (2008). A Brief Introduction to Fourier Analysis on the Boolean Cube. *Theory of Computing ToC Library, Graduate Surveys 1*.

Manuscript received 19 June 2013

ANDRIS AMBAINIS  
Faculty of Computing  
University of Latvia  
Raiņa bulv. 19, Rīga, LV-1586  
Latvia  
ambainis@lu.lv

RONALD DE WOLF  
CWI and University of Amsterdam  
P.O. Box 94079  
NL-1090 GB Amsterdam  
The Netherlands  
rdewolf@cwi.nl