Lower Bounds on Quantum Query Complexity

Peter Høyer* hoyer@cpsc.ucalgary.ca

Robert Špalek† sr@cwi.nl

September 2005

Abstract

Shor's and Grover's famous quantum algorithms for factoring and searching show that quantum computers can solve certain computational problems significantly faster than any classical computer. We discuss here what quantum computers *cannot* do, and specifically how to prove limits on their computational power. We cover the main known techniques for proving lower bounds, and exemplify and compare the methods.

1 Introduction

The very first issue of the Journal of the ACM was published in January 1954. It was the first journal devoted to computer science. For its 50th anniversary volume, published in January 2003, editors-in-chief Joseph Y. Halpern asked winners of the Turing Award and the Nevanlinna Prize to discuss up to three problems that they thought would be major problems for computer science in the next 50 years. Nevanlinna Prize winner Leslie G. Valiant [Val03] describes three problems, the first of which is on physically realizable models for computation and formalizes the setting by defining: "We therefore call our class PhP, the class of physically constructible polynomial resource computers." He then formulates the problem by: "[t]o phrase a single question, the full characterization of PhP," and argues that "this single question appears at this time to be scientifically the most fundamental in computer science."

On January 26, this year, Nobel Laureate David Gross gave a CERN Colloquium presentation on "The future of physics" [Gro05]. He discusses "25 questions that might guide physics, in the broadest sense, over the next 25 years," and includes as questions 15 and 16 "Complexity" and "Quantum Computing." In July, this year, the Science magazine celebrated its 125th anniversary by "explor[ing] 125 big questions that face scientific enquiry over the next quarter-century" [Sei05].

*Department of Computer Science, University of Calgary. Supported by Canada's Natural Sciences and Engineering Research Council (NSERC), the Canadian Institute for Advanced Research (CIAR), and The Mathematics of Information Technology and Complex Systems (MITACS).

†CWI and University of Amsterdam. Supported in part by the EU fifth framework project RESQ, IST-2001-37559. Work conducted in part while visiting the University of Calgary.

Among the top 25, is the question of "What are the limits of conventional computing?" Charles Seife writes: "[T]here is a realm beyond the classical computer: the quantum," and he discusses the issue of determining "what quantum-mechanical properties make quantum computers so powerful."

In this issue of the Bulletin of the EATCS, we would like to offer an introduction to the topic of studying limitations on the power of quantum computers. Can quantum computers really be more powerful than traditional computers? What can quantum computers not do? What proof techniques are used for proving bounds on the computational power of quantum computers? It is a highly active area of research and flourishing with profound and beautiful theorems. Though deep, it is fortunately also an accessible area, based on basic principles and simple concepts, and one that does not require specialized prior knowledge. One aim of this paper is to show this by providing a fairly complete introduction to the two most successful methods for proving lower bounds on quantum computations, the adversary method and the polynomial method. Our survey is biased towards the adversary method since it is likely the least familiar method and it yields very strong lower bounds. This paper is meant to be supplemented by the excellent survey of Buhrman and de Wolf [BW02] on decision tree complexities, published in 2002 in the journal Theoretical Computer Science.

We demonstrate the methods on a running example, and for this, we use one of the most basic algorithmic questions one may think of: that of searching an ordered set. Can one implement ordered searching significantly faster on a quantum computer than applying a standard $\Theta(\log N)$ binary search algorithm?

The rest of the paper is organized as follows. We motivate and define our models of computation in the next section. We then discuss very basic principles used in proving quantum lower bounds in Section 3 and use them to establish our first lower-bound method, the adversary method, in Section 4. We discuss how to apply the method in Section 5, and its limitations in Section 6. We then give an introduction to the second method, the polynomial method, in Section 7. We compare the two methods in Section 8 and give a few final remarks in Section 9.

We have aimed at limiting prior knowledge on quantum computing to a bare minimum. Sentences and paragraphs with kets and bras ($|$this is a ket$\rangle$ and $\langle$this is a bra$|$) can either safely be skipped, or substituted with column-vectors and row-vectors, respectively.

2 Quantum query complexity

Many quantum algorithms are developed for the so-called oracle model in which the input is given as an oracle so that the only knowledge we can gain about the input is in asking queries to the oracle. The input is a finite bitstring $x \in \{0,1\}^N$ of some length N, where $x = x_1 x_2 \dots x_N$. The goal is to compute some function $F : \{0,1\}^N \to \{0,1\}^m$ of the input x. Some of the functions we consider are boolean, some not. We use the shorthand notation $[N] = \{1, 2, \dots, N\}$.

As our measure of complexity, we use the query complexity. The query complexity of an algorithm A computing a function F is the number of queries used by A. The query complexity of F is the minimum query complexity of any algorithm computing F. We are interested in proving lower bounds on the query complexity of specific functions and consider methods for computing

such lower bounds.

An alternative measure of complexity would be to use the time complexity which counts the number of basic operations used by an algorithm. The time complexity is always at least as large as the query complexity since each query takes one unit step, and thus a lower bound on the query complexity is also a lower bound on the time complexity. For most existing quantum algorithms, including Grover's algorithm [Gro96], the time complexity is within poly-logarithmic factors of the query complexity. A notorious exception is the so-called Hidden Subgroup Problem which has polynomial query complexity [EHK04], yet polynomial time algorithms are known only for some instances of the problem.

The oracle model is called decision trees in the classical setting. A classical query consists of an index $i \in [N]$, and the answer of the bit x_i. There is a natural way of modelling a query so that it is reversible. The input is a pair (i, b), where $i \in [N]$ is an index and $b \in \{0, 1\}$ a bit. The output is the pair $(i, b \oplus x_i)$, where the bit b is flipped if $x_i = 1$. There are (at least) two natural ways of generalizing a query to the quantum setting, in which we require all operations to be unitary. The first way is to consider a quantum query as a unitary operator that takes two inputs $|i\rangle|b\rangle$, where $i \in [N]$ and $b \in \{0, 1\}$, and outputs $|i\rangle|b \oplus x_i\rangle$. The oracle is then simply just a linear extension of the reversible query given above. We extend the definition of the oracle so that we can simulate a non-query, and we allow it to take some arbitrary ancilla state $|z\rangle$ with $z \geq 0$ as part of the input and that is acted upon trivially,

$$\mathsf{O}'_x|i, b; z\rangle = \begin{cases} |i, b; z\rangle & \text{if } i = 0 \text{ or } x_i = 0 \\ |i, b \oplus 1; z\rangle & \text{if } i \in [N] \text{ and } x_i = 1. \end{cases} \tag{1}$$

The ancilla $|z\rangle$ contains any additional information currently part of the quantum state that is not involved in the query.

The second way is to consider a quantum query as a unitary operator O_x that takes only the one input $|i\rangle$ and outputs $(-1)^{x_i}|i\rangle$, where $i \in [N]$. We say that the oracle is "computed in the phases" by O_x. Both operators O'_x and O_x square to the identity, i.e., they are their own inverses, and thus unitary. The two operators are equivalent in that one query to either oracle can be simulated by a superposition query to the other oracle preceeded and followed by a basis change. Though the first way is possibly the more intuitive, we shall adapt the second way as it is very convenient when proving lower bounds. Again, we extend the definition of the oracle O_x so that it also embodies a non-query, and we allow it to take some arbitrary ancilla state $|z\rangle$ that is not acted upon,

$$\mathsf{O}_x|i; z\rangle = \begin{cases} |i; z\rangle & \text{if } i = 0 \\ (-1)^{x_i}|i; z\rangle & \text{if } 1 \leq i \leq N. \end{cases} \tag{2}$$

We may think of one query as a one-round exchange of information between two parties, the algorithm and the oracle. In the classical setting, the algorithm sends an index $i \in [N]$ to the oracle, and the oracle responds with one bit of information, namely x_i. In the quantum setting, the algorithm sends the $\log_2(N)$ qubits $|i\rangle$ to the oracle O_x, and the oracle responds with $(-1)^{x_i}|i\rangle$. The algorithm and oracle thus exchange a total number of $2\log_2(N)$ qubits, and thus, a quantum query to O_x can convey up to $2\log_2(N)$ classical bits of information about the oracle by Holevo's theorem [Hol73, CD$^+$98] and superdense coding [BW92].

Information theoretically, a function $F : \{0,1\}^N \to \{0,1\}^{\log_2(N)}$ that outputs at most $O(\log_2(N))$ bits, can potentially be solved by a constant number of queries to the oracle. An example of such a problem is the Deutsch-Jozsa problem [DJ92], which is to distinguish balanced boolean functions from constant functions. (A function F is constant if $F(x) = F(y)$ for all inputs x, y, and it is balanced if it is not constant and $|F^{-1}(F(x))| = |F^{-1}(F(y))|$ for all inputs x, y.)

A quantum algorithm in the oracle model starts in a state that is independent of the oracle. For convenience, we choose the state $|0\rangle$ in which all qubits are initialized to 0. It then evolves by applying arbitrary unitary operators U to the system, alternated with queries O_x to the oracle x, followed by a conclusive measurement of the final state, the outcome of which is the result of the computation. In symbols, a quantum algorithm A that uses T queries, computes the final state

$$|\psi_x^T\rangle = \mathsf{U}_T \mathsf{O}_x \mathsf{U}_{T-1} \cdots \mathsf{U}_1 \mathsf{O}_x \mathsf{U}_0 |0\rangle \tag{3}$$

which is then measured. If the algorithm computes some function $F : \{0,1\}^N \to \{0,1\}^m$, we measure the m leftmost bit of the final state $|\psi_x^T\rangle$, producing some outcome w. The success probability p_x of A on input $x \in \{0,1\}^N$ is the probability that $w = F(x)$. For complete functions $F : \{0,1\}^N \to \{0,1\}^m$, we define the success probability of A as the minimum of p_x over all $x \in \{0,1\}^N$. For partial functions $F : S \to \{0,1\}^m$, where $S \subseteq \{0,1\}^N$, we take the minimum over S only. A quantum algorithm A has error at most ϵ if the success probability of A is at least $1 - \epsilon$. Let $Q_\epsilon(F)$ denote the minimum query complexity of any quantum algorithm that computes F with two-sided error at most ϵ, and as common, let $Q_2(F) = Q_{1/3}(F)$ denote the two-sided bounded error complexity with $\epsilon = 1/3$.

As our running example, we use the well-known ordered searching problem. In the oracle model, the input to ordered searching is an N-bit string $x = (x_1, \ldots, x_N)$. We are promised that $x_i \leq x_{i+1}$ for all $1 \leq i < N$ and that $x_N = 1$, and the goal is to find the leftmost 1, i.e., the index $i \in [N]$ for which $x_i = 1$ and no index $j < i$ exists with $x_j = 1$.

Given: An N-bit string $x = (x_1, x_2, \ldots, x_N)$ given as an oracle.

Promise: $x_i \leq x_{i+1}$ for $1 \leq i < N$ and $x_N = 1$.

Output: Index i such that $x_i = 1$ and either $x_{i-1} = 0$ or $i = 1$.

The classical query complexity of ordered searching is $\lceil \log_2(N) \rceil$ and is achieved by standard binary searching. The quantum query complexity is at most $0.45 \log_2 N$, due to the work of high school student M. B. Jacokes in collaboration with Landahl and Brookes [JLB05] (See also [FGGS, HNS02]). Using the adversary method, we show that their algorithm is within a factor of about two of being optimal.

3 Distinguishing hard inputs

The first quantum lower bound using adversary arguments was given by Bennett, Bernstein, Brassard, and Vazirani in [BB$^+$97]. They show that any quantum query algorithm can be sensitive to at most quadratically many oracle bits, which implies a lower bound of $\Omega(\sqrt{N})$ for Grover's

problem [Gro96] and thus proves that Grover's $O(\sqrt{N})$ algorithm is optimal. Grover's problem is a search problem in which we are given an N-bit string $x \in \{0,1\}^N$ as an oracle, and the goal is to find an index i for which $x_i = 1$, provided one exists. Interestingly, the lower bound of Bennett et al. was proved in 1994, well before Grover defined his search problem. In 2000, Ambainis [Amb02] found an important generalization of the method and coined it "adversary arguments."

A constructive interpretation of basic adversary arguments is in terms of *distinguishability*. We will thus not be concerned with computing the function F, but merely interested in distinguishing oracles. Consider some algorithm A that computes some function F in the oracle model, and consider two inputs $x, y \in \{0,1\}^N$ for which $F(x) \neq F(y)$. Since A computes F, it must in particular be capable of distinguishing between oracle x and oracle y. For a given problem we try to identify *pairs of oracles* that are hard to *distinguish*. If we can identify hard input pairs, we may derive a good lower bound. However, a caveat is that using only the very hardest input pairs does not yield good lower bounds for some problems, and we are thus naturally led to also consider less hard input pairs. A remedy is to use *weights* that capture the hardness of distinguishing each pair of oracles, and to do so, we define a matrix Γ of dimension $2^N \times 2^N$ that takes non-negative real values,

$$\Gamma : \{0,1\}^N \times \{0,1\}^N \to \Re_0^+. \tag{4}$$

We require that Γ is symmetric and that $\Gamma[x,y] = 0$ whenever $F(x) = F(y)$. We say that Γ is a *spectral adversary matrix for* F if it satisfies these two conditions. The symmetry condition on Γ states that we are concerned with distinguishing *between* any two inputs x, y. We are not concerned with distinguishing x *from* y, nor distinguishing y *from* x. We discuss this subtlety further in Section 5 below when considering alternative definitions of weighted adversary arguments. The spectral adversary matrix Γ allows us to capture both total and partial functions, as well as non-boolean functions. Since we are only concerned with distinguishability, once we have specified the entries of Γ, we may safely ignore the underlying function F.

Weighted adversary arguments were first used by Høyer, Neerbek, and Shi in [HNS02] to prove a lower bound of $\Omega(\log N)$ for ordered searching and $\Omega(N \log N)$ for sorting. Barnum and Saks [BS04] used weighted adversary arguments to prove a lower bound of $\Omega(\sqrt{N})$ for read-once formulae, and introduced the notion Γ that we adapt here. Barnum, Saks, and Szegedy extended their work in [BSS03] and derived a general lower bound on the query complexity of F in terms of spectral properties of matrix Γ. Their lower bound has a very elegant and short formulation, a basic proof, and captures important properties of adversary methods, and we shall thus adapt much of their terminology.

As discussed above, the key to prove a good lower bound is to pick a good adversary matrix Γ. For our running example of ordered searching, which is a partial non-boolean function, we use the following weights.

Example: Ordered Seaching 1 *The weight on the pair (x,y) is the inverse of the Hamming distance of x and y,*

$$\Gamma^{\text{search}}[x,y] = \begin{cases} \frac{1}{|F(x)-F(y)|} & \text{if } x \text{ and } y \text{ are valid and distinct inputs to } F \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The larger the Hamming distance between x and y, the easier it is to distinguish them, and the smaller weight is assigned to the pair.

We have to choose how to measure distinguishability. The possibly simplest measure is to use inner products. Two quantum states are distinguishable with certainty if and only if they are orthogonal, and they can be distinguished with high probability if and only if their inner product has small absolute value.

Fact 1 *Suppose we are given one of two known states $|\Psi_x\rangle, |\Psi_y\rangle$. There exists a measurement that correctly determines which of the two states we are given with error probability at most ϵ if and only if $|\langle\Psi_x|\Psi_y\rangle| \leq \epsilon'$, where $\epsilon' = 2\sqrt{\epsilon(1-\epsilon)}$.*

Since a unitary operator is just a change of basis, it does not change the inner product between any two quantum states, and thus the inner product can only change as a consequence of queries to the oracle.

4 Adversary lower bounds

Adversary lower bounds are of information theoretical nature. A basic idea in adversary lower bounds is to upper-bound the amount of information that can be learned in a single query. If little information can be learned in any one query, then many queries are required. We use spectral properties of Γ to put an upper bound on the amount of information the algorithm learns about the oracle.

Let A be some quantum algorithm that computes some function F with bounded two-sided error. For every integer $t \geq 0$ and every oracle x, let

$$|\psi_x^t\rangle = \mathsf{U}_t\mathsf{O}_x \cdots \mathsf{U}_1\mathsf{O}_x\mathsf{U}_0|0\rangle \tag{6}$$

denote the quantum state after t queries to the oracle. To measure the progress of the algorithm, we define similarly to [Amb02, HNS02, BS04, BSS03] a weight function

$$W^t = \sum_{x,y} \Gamma[x,y]\delta_x\delta_y \cdot \langle\psi_x^t|\psi_y^t\rangle, \tag{7}$$

where δ is a fixed principal eigenvector of Γ, i.e., a normalized eigenvector corresponding to the largest eigenvalue of Γ, and where δ_x denotes the x^{th} entry of δ.

The algorithm starts in a quantum state $|\psi_x^0\rangle = \mathsf{U}_0|0\rangle$ which is independent of the oracle x, and thus the total initial weight is

$$W^0 = \sum_{x,y} \Gamma[x,y]\delta_x\delta_y = \lambda(\Gamma), \tag{8}$$

where $\lambda(\Gamma)$ denotes the spectral norm of Γ. The final state of the algorithm after T queries is $|\psi_x^T\rangle$ if the oracle is x, and it is $|\psi_y^T\rangle$ if the oracle is y. If $F(x) \neq F(y)$, we must have that $|\langle\psi_x^T|\psi_y^T\rangle| \leq \epsilon'$

by Fact 1, and hence $W^T \leq \epsilon' W^0$. If the total weight can decrease by at most Δ by each query, the algorithm requires $\Omega(\frac{W^0}{\Delta})$ queries to the oracle.

Following Barnum, Saks, and Szegedy [BSS03], we upper bound Δ by the largest spectral norm of the matrices Γ_i, defined by

$$\Gamma_i[x,y] = \begin{cases} \Gamma[x,y] & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i, \end{cases} \tag{9}$$

for each $1 \leq i \leq n$. The theorem of [BSS03] is here stated (and proved) in a slightly more general form than in [BSS03] so that it also applies to non-boolean functions. Our proof aims at emphasizing distinguishability and differs from the original.

Theorem 2 (Spectral method [BSS03]) *For any adversary matrix* Γ *for any function* $F : \{0,1\}^N \to \{0,1\}^m$,

$$Q_2(F) = \Omega\Big(\frac{\lambda(\Gamma)}{\max_i \lambda(\Gamma_i)}\Big). \tag{10}$$

Proof We prove that the drop in total weight $W^t - W^{t+1}$ by the $t+1^{\text{st}}$ query is upper-bounded by the largest eigenvalue of the matrices Γ_i.

For each $0 \leq i \leq N$, let $\mathsf{P}_i = \sum_{z \geq 0} |i;z\rangle\langle i;z|$ denote the projection onto the subspace querying the i^{th} oracle bit. Let $\beta_{x,i} = |\mathsf{P}_i|\psi_x^t\rangle|$ denote the absolute value of the amplitude of querying the i^{th} bit in the $t+1^{\text{st}}$ query, provided the oracle is x. Note that $\sum_{i=0}^N \beta_{x,i}^2 = 1$ for any oracle x, since the algorithm queries one of the N bits $x_1, \ldots, x_N$, or simulates a non-query by querying the oracle with $i = 0$. The $t+1^{\text{st}}$ query changes the inner product by at most the overlap between the projections of the two states onto the subspace that corresponds to indices i on which x_i and y_i differ,

$$\Big|\langle\psi_x^t|\psi_y^t\rangle - \langle\psi_x^{t+1}|\psi_y^{t+1}\rangle\Big| = \Big|\langle\psi_x^t|(\mathsf{I} - \mathsf{O}_x\mathsf{O}_y)|\psi_y^t\rangle\Big| = \Big|2\sum_{i:x_i\neq y_i}\langle\psi_x^t|\mathsf{P}_i|\psi_y^t\rangle\Big| \leq 2\sum_{i:x_i\neq y_i}\beta_{x,i}\beta_{y,i}. \tag{11}$$

The bigger the amplitudes of querying the bits i on which x_i and y_i differ, the larger the drop in the inner product can be.

Define an auxiliary vector $a_i[x] = \delta_x\beta_{x,i}$ and note that

$$\sum_{i=0}^N |a_i|^2 = \sum_{i=0}^N\sum_x \delta_x^2\beta_{x,i}^2 = \sum_x \delta_x^2 \sum_{i=0}^N \beta_{x,i}^2 = \sum_x \delta_x^2 = 1.$$

The drop in the total weight is upper-bounded by

$$\begin{aligned}
\left|W^t - W^{t+1}\right| &= \Big|\sum_{x,y}\Gamma[x,y]\delta_x\delta_y\big(\langle\psi_x|\psi_y\rangle - \langle\psi'_x|\psi'_y\rangle\big)\Big| \\
&= \Big|2\sum_{x,y}\sum_{i:x_i\neq y_i}\Gamma[x,y]\delta_x\delta_y\langle\psi_x|\mathsf{P}_i|\psi_y\rangle\Big| \\
&\leq 2\sum_{x,y}\sum_{i}\Gamma_i[x,y]\delta_x\delta_y\cdot\beta_{x,i}\beta_{y,i} \\
&= 2\sum_i a_i^*\Gamma_i a_i \\
&\leq 2\sum_i \lambda(\Gamma_i)|a_i|^2 \\
&\leq 2\max_i\lambda(\Gamma_i)\cdot\sum_i|a_i|^2 \\
&= 2\max_i\lambda(\Gamma_i).
\end{aligned}$$

Here a_i^* denotes the transpose of a_i. The first inequality bounds the drop in inner product for a specific pair and follows from Equation 11. The second inequality follows from the spectral norm of Γ. The second and third inequalities state that the best possible query distributes the amplitude of the query according to the largest principal eigenvector of the query matrices Γ_i. □

Example: Ordered Seaching 2 *Returning to our example of ordered searching, for $N = 4$, the adversary matrix with respect to the ordered basis $(0001, 0011, 0111, 1111)$ is given by*

$$\Gamma^{\text{search}(4)} = \begin{bmatrix} 0 & 1 & \frac{1}{2} & \frac{1}{3} \\ 1 & 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & 0 & 1 \\ \frac{1}{3} & \frac{1}{2} & 1 & 0 \end{bmatrix}.$$

The spectral norm is easily seen to be lower-bounded by the sum of the entries in the first row, $\lambda(\Gamma^{\text{search}(4)}) \geq 1 + \frac{1}{2} + \frac{1}{3}$. In general, $\lambda(\Gamma^{\text{search}})$ is lower-bounded by the harmonic number H_{N-1}, which is at least $\ln(N)$. The spectral norm of the query matrices $\lambda(\Gamma_i^{\text{search}})$ is maximized when $i = \lfloor N/2 \rfloor$, in which case it is upper-bounded by the spectral norm of the infinite Hilbert matrix $[1/(r+s-1)]_{r,s\geq 1}$, which is π. We thus reprove the lower bound of $(1-\epsilon')\frac{\ln(N)}{\pi}$ for ordered searching given in [HNS02].

5 Applying the spectral method

The spectral method is very appealing in that it has a simple formulation, a basic proof, and gives good lower bounds for many problems. Špalek and Szegedy [ŠS05] show that for any problem, the best lower bound achievable by the spectral method is always at least as good as the best

lower bound achievable by any of the previously published adversary methods. Their proof is constructive and illuminating: given any lower bound in any of the previously published adversary methods, they construct an adversary matrix Γ and prove it achieves the same lower bound.

The first general quantum lower bound using adversary arguments was introduced by Ambainis in [Amb02]. As shown in [ŠS05], it can be derived from the spectral method by applying simple bounds on the spectral norm of Γ and each Γ_i. By definition, the numerator $\lambda(\Gamma)$ is lower-bounded by $\frac{1}{|d|^2} d^* \Gamma d$ for any non-negative vector d, and by Mathias' lemma [Mat90], the denominator $\lambda(\Gamma_i)$ is upper-bounded by the product of a row-norm and a column-norm.

Lemma 3 ([Mat90, ŠS05]) *Let G be any non-negative symmetric matrix and M, N non-negative matrices such that $G = M \circ N$ is the entrywise product of M and N. Then*

$$\lambda(G) \leq \max_{\substack{x,y \\ G[x,y]>0}} r_x(M)\, c_y(N),$$

where $r_x(M)$ is the ℓ_2-norm of the x^{th} row in M, and $c_y(N)$ is the ℓ_2-norm of the y^{th} column in N.

Applying these two bounds, we obtain Ambainis' lower bound in [Amb02]. We refer to the method as an unweighted adversary method since it considers only two types of inputs: easy inputs and hard inputs. We construct a zero-one valued adversary matrix Γ that corresponds to a uniform distribution over the hard input pairs.

Theorem 4 (Unweighted method [Amb02]) *Let F be a partial boolean function, and let $A \subseteq F^{-1}(0)$ and $B \subseteq F^{-1}(1)$ be subsets of (hard) inputs. Let $R \subseteq A \times B$ be a relation, and set $R_i = \{(x,y) \in R : x_i \neq y_i\}$ for each $1 \leq i \leq n$. Let m, m' denote the minimal number of ones in any row and any column in relation R, respectively, and let ℓ, ℓ' denote the maximal number of ones in any row and any column in any of the relations R_i, respectively. Then $Q_2(f) = \Omega(\sqrt{mm'/\ell\ell'})$.*

Proof Let $S = \{(x,y) : (x,y) \in R \vee (y,x) \in R\}$ be a symmetrized version of R. Define a column vector d from the relation S by setting $d_x = \sqrt{|\{y : (x,y) \in S\}|}$, and an adversary matrix Γ by setting $\Gamma[x,y] = \frac{1}{d_x d_y}$ if and only if $(x,y) \in S$. Then $\lambda(\Gamma) \geq \frac{1}{|d|^2} d^* \Gamma d = 1$. For each of the matrices Γ_i, we apply Lemma 3 with $M[x,y] = N[y,x] = \frac{1}{d_x}$ if and only if $(x,y) \in S$. For every $(x,y) \in R$, $r_x(M) \leq \sqrt{\ell/d_x^2} \leq \sqrt{\ell/m}$ and $c_y(N) \leq \sqrt{\ell'/d_y^2} \leq \sqrt{\ell'/m'}$. For every $(x,y) \in S - R$, the two inequalities are swapped. By Lemma 3, $\lambda(\Gamma_i) \leq \max_{x,y:\Gamma_i[x,y]>0} r_x(M) c_y(N) \leq \sqrt{\ell\ell'/mm'}$. □

The unweighted adversary method is very simple to apply as it requires only to specify a set R of hard input pairs. It gives tight lower bounds for many computational problems, including inverting a permutation [Amb02], computing any symmetric function and counting [NW99, BC$^+$99, BH$^+$02], constant-level and-or trees [Amb02, HMW03], and various graph problems [DH$^+$04]. For some computational problems, the hardness does however not necessarily rely only on a few selected hard instances, but rather on more global properties of the inputs. Applying the unweighted method on ordered searching would for instance only yield a lower bound of a constant. In these cases, we may apply the following weighted variant of the method, due to Ambainis [Amb03] and Zhang [Zha04].

Theorem 5 (Weighted method [Amb03, Zha04]) *Let $F : S \to \{0,1\}^m$ be a partial function. Let w, w' denote a weight scheme as follows:*

- *Every pair $(x,y) \in S^2$ is assigned a non-negative weight $w(x,y) = w(y,x)$ that satisfies $w(x,y) = 0$ whenever $F(x) = F(y)$.*
- *Every triple $(x,y,i) \in S^2 \times [N]$ is assigned a non-negative weight $w'(x,y,i)$ that satisfies $w'(x,y,i) = 0$ whenever $x_i = y_i$ or $F(x) = F(y)$, and $w'(x,y,i)w'(y,x,i) \geq w^2(x,y)$ for all x, y, i with $x_i \neq y_i$.*

Then

$$Q_2(F) = \Omega\left(\min_{\substack{x,y,i \\ w(x,y)>0 \\ x_i \neq y_i}} \sqrt{\frac{wt(x)wt(y)}{v(x,i)v(y,i)}} \right),$$

where $wt(x) = \sum_y w(x,y)$ and $v(x,i) = \sum_y w'(x,y,i)$ for all $x \in S$ and $i \in [N]$.

At first glance, the weighted method may look rather complicated, both in its formulation and use, though it is not. We first assign weights to pairs (x,y) of inputs for which $F(x) \neq F(y)$, as in the spectral method. We require the weights to be symmetric so that they represent the difficulty in distinguishing *between* x *and* y.

We then afterwards assign weights $w'(x,y,i)$ that represent the difficulty in distinguishing x *from* y *by querying index* i. The harder it is to distinguish x from y by index i, compared to distinguishing y from x by index i, the more weight we put on (x,y,i) and the less on (y,x,i), and vice versa.

To quantify this, define $t(x,y,i) = w'(x,y,i)/w'(y,x,i)$. Then $t(x,y,i)$ represents the relative amount of information we learn about input pairs (x,z) compared to the amount of information we learn about input pairs (u,y), by querying index i. If we, by querying index i, learn little about x compared to y, we let $t(x,y,i)$ be large, and otherwise small. Consider we query an index i for which $x_i \neq y_i$. Then we learn whether the oracle is x or y. However, at the same time, we also learn whether the oracle is x or z for any other pair (x,z) for which $x_i \neq z_i$ and $F(x) \neq F(z)$; and similarly, we learn whether the oracle is u or y for any other pair (u,y) for which $u_i \neq y_i$ and $F(u) \neq F(y)$. The less information querying index i provides about pairs (x,z) compared to pairs (u,y), the larger we choose $t(x,y,i)$. Having thus chosen $t(x,y,i)$, we set $w'(x,y,i) = w(x,y)\sqrt{t(x,y,i)}$ and $w'(y,x,i) = w(x,y)/\sqrt{t(x,y,i)}$.

We show next that the weighted method yields a lower bound of $\Omega(\log N)$ for the ordered searching problem. This proves that the weighted method is strictly stronger than the unweighted method. The weighted method yields strong lower bounds for read-once formula [BS04] and iterated functions [Amb03]. Aaronson [Aar04], Santha and Szegedy [SS04], and Zhang [Zha05] use adversary arguments to prove lower bounds for local search, a distributed version of Grover's problem. Špalek and Szegedy prove in [ŠS05] that the weighted method is equivalent to the spectral method—any lower bound that can be achieved by one of the two methods can also be shown by the other. Their proof is constructive and gives simple expressions for converting one into the other. The main weights $w(x,y)$ are the coefficients of the weight function W^t for the input pair (x,y), that is, $w(x,y) = \Gamma[x,y]\delta_x\delta_y$, and the secondary weights $w'(x,y,i)$ follow from Mathias' lemma [Mat90] (Lemma 3).

Example: Ordered Seaching 3 *To apply the weighted method on ordered searching, we pick the same weights* $w(x,y) = \Gamma^{\text{search}}[x,y]\,\delta_x\delta_y$ *as in the spectral method as there are no strong reasons for choosing otherwise. Now, consider* $t(x,y,i)$ *with* $F(x) \leq i < F(y)$ *so that* $x_i \neq y_i$*. By querying index* i*, we also learn to distinguish between* x *and* z *for each of the* $F(y) - i$ *inputs* z *with* $i < F(z) \leq F(y)$*, and we learn to distinguish between* u *and* y *for each of the* $i - F(x) + 1$ *inputs* u *with* $F(x) \leq F(u) \leq i$*. We thus choose to set*

$$t(x,y,i) = \frac{|F(x) - i| + 1}{|F(y) - i| + 1}.$$

Plugging these values into the weighted method yields a lower bound of $\Omega(\log N)$ *for ordered searching.*

6 Limitations of the spectral method

The spectral method and the weighted adversary method bound the amount of information that can be learned in any one query. They do not take into account that the amount of information that can be learned in the j^{th} query might differ from the amount of information that can be learned in the k^{th} query.

In 1999, Zalka [Zal99] successfully managed to capture the amount of information that can be learned in each individual query for a restricted version of Grover's problem [Gro96]. In this restricted version, we are promised that the input oracle x is either the zero-string (so $|x| = 0$) or exactly one entry in x is one (so $|x| = 1$), and the goal is to determine which is the case. By symmetry considerations, Zalka demonstrates that Grover's algorithm saturates some improved inequalities (which are similar to Eq. 11) and hence is optimal, even to within an additive constant.

Since current adversary methods do not capture the amount of information the algorithm currently knows, we may simply assume that the algorithm already knows every bit of the oracle and that it tries to prove so. This motivates a study of the relationship between the best bound achievable by the spectral method and the certificate complexity. A *certificate* for an input $x \in \{0,1\}^N$, is a subset $C \subseteq [N]$ of input bits such that for any other input y in the domain of F that may be obtained from x by flipping some of the indices not in C, we have that $F(x) = F(y)$. The certificate complexity $\mathrm{C}_x(F)$ of input x is the size of a smallest certificate for x. The *certificate complexity* $\mathrm{C}(F)$ of a function F is the maximum certificate complexity of any of its inputs. We also define the z-certificate complexity $\mathrm{C}_z(F)$ when taking the maximum only over inputs that map to z. The spectral theorem can then never yield a lower bound better than a quantity that can be expressed in terms of certificate complexity.

Lemma 6 ([LM04, Zha04, ŠS05]) *Let* $F : S \to \{0,1\}$ *be any partial boolean function. The spectral adversary lower bound* $\mathrm{Adv}(F)$ *is at most* $\min\left\{\sqrt{\mathrm{C}_0(F)N}, \sqrt{\mathrm{C}_1(F)N}\right\}$*. If* F *is total, the method is limited by* $\sqrt{\mathrm{C}_0(F)\mathrm{C}_1(F)}$*.*

The certificate complexity of a function $F : \{0,1\}^N \to \{0,1\}^m$ is itself polynomially related to the block sensitivity of the function. An input $x \in \{0,1\}^N$ is *sensitive* to a block $B \subseteq [N]$

if $F(x) \neq F(x^B)$, where x^B denotes the input obtained by flipping the bits in x with indices from B. The block sensitivity $\mathrm{bs}_x(F)$ of input x is the maximum number of disjoint blocks $B_1, B_2, \ldots, B_k \subseteq [N]$ on which x is sensitive. The *block sensitivity* $\mathrm{bs}(F)$ of F is the maximum block sensitivity of any of its inputs. We also define the z-block sensitivity $\mathrm{bs}_z(F)$ when taking the maximum only over inputs that map to z.

For any boolean function $F : \{0,1\}^N \to \{0,1\}$, the certificate complexity is upper-bounded by $\mathrm{C}(F) \leq \mathrm{bs}_0(F)\mathrm{bs}_1(F)$, and thus so is the spectral adversary method. Conversely, $\mathrm{Adv}(F) \geq \sqrt{\mathrm{bs}(F)}$ by a zero-one valued adversary matrix Γ: Let $x' \in \{0,1\}^N$ be an input that achieves the block sensitivity of F, and let $B_1, B_2, \ldots, B_k \subseteq [N]$ be disjoint blocks on which x' is sensitive, where $k = \mathrm{bs}(F)$. Set $\Gamma(F)[x, x^B] = 1$ if and only if $x = x'$ and B is one of the k blocks B_i and close Γ under transposition. Then $\lambda(\Gamma) = \sqrt{k}$ and $\max_i \lambda(\Gamma_i) = 1$, and thus

$$\sqrt{\mathrm{bs}(F)} \leq \mathrm{Adv}(F) \leq \mathrm{bs}_0(F)\mathrm{bs}_1(F). \tag{12}$$

The spectral adversary method is not suitable for proving lower bounds for problems related to property testing. If function $F : S \to \{0,1\}$ is a partial function with $S \subseteq \{0,1\}^N$ such that every zero-input is of Hamming distance at least εn from every one-input, then the spectral theorem does not yield a lower bound better than $1/\varepsilon$.

Laplante and Magniez introduce in [LM04] a lower-bound method based on Kolmogorov complexity. They show by direct constructions that their method is at least as strong as each of the two methods, the spectral and weighted adversary method. Špalek and Szegedy then show in [ŠS05] that the spectral method is at least as strong as the Kolmogorov complexity method, allowing us to conclude that the three methods are equivalent. Having such a variety of representations of the same method shows that the adversary method is very versatile and captures fundamental properties of functions. Indeed, Laplante, Lee, and Szegedy [LLS05] show that the square of the adversary bound is a lower bound on the formula size. The following lower-bound method is a combinatorial version of the Kolmogorov complexity method.

Theorem 7 (Minimax method [LM04, ŠS05]) *Let $F : S \to \{0,1\}^m$ be a partial function and* A *a bounded-error quantum algorithm for F. Let $p : S \times [N] \to \Re_0^+$ be a set of $|S|$ probability distributions such that $p_x(i)$ denotes the average probability of querying the i^{th} input bit on input x, where the average is taken over the whole computation of* A*. Then the query complexity Q_{A} of algorithm* A *satisfies*

$$Q_{\mathsf{A}} \geq M_p = \max_{x,y:F(x)\neq F(y)} \frac{1}{\sum_{i:x_i\neq y_i} \sqrt{p_x(i)\, p_y(i)}}.$$

The previous methods satisfy the property that if we plug in some matrix or relation, we get a valid lower bound. The minimax method is principally different. A lower bound computed by the minimax theorem holds for one particular algorithm A, and it may not hold for some other and better algorithm. However, we may obtain a universal lower bound that holds for *every* bounded error algorithm by simply taking the minimum of the bound M_p over all possible sets of probability distributions p. The spectral bound and the minimax bound are in a primal-dual relation: the best lower bound that can be obtained by any adversary matrix Γ equals the smallest bound that can

be obtained by a set of probability distributions p [ŠS05]. Primal methods are used for obtaining concrete lower bounds and dual methods are used for proving limitations of the method, as in Lemma 6.

A useful property of the adversary method is that it composes. Consider a function of the form $H = F \circ (G_1, \ldots, G_k)$, where $F : \{0,1\}^k \to \{0,1\}$ and $G_i : \{0,1\}^{N_i} \to \{0,1\}$ for $i = 1, \ldots, k$ are partial boolean functions. A composition theorem states the complexity of function H in terms of the complexities of F and $G_1, \ldots, G_k$. Barnum and Saks [BS04] use composition properties to prove a query lower bound of $\Omega(\sqrt{N})$ for any read-once formula, Ambainis [Amb03] proves a composition lower bound for iterated boolean functions, and Laplante, Lee, and Szegedy [LLS05] prove a limitation on composition lower bounds for functions G_i for which the adversary bound is upper bounded by a common bound b. To formulate a composition theorem for arbitrary cases when the functions G_i may have different adversary bounds, we require a weighted version of the spectral method.

Let $F : \{0,1\}^N \to \{0,1\}$ be a partial boolean function and $\alpha = (\alpha_1, \ldots, \alpha_N)$ a string of positive reals. Let

$$\mathrm{Adv}_\alpha(F) = \max_\Gamma \min_i \left\{ \alpha_i \frac{\lambda(\Gamma)}{\lambda(\Gamma_i)} \right\},$$

where Γ ranges over all adversary matrices for F. If the weights are all 1, then our new quantity $\mathrm{Adv}_\alpha(F)$ coincides with the spectral adversary bound and is thus a lower bound on the quantum query complexity of F. If the weights α are non-uniform, then $\mathrm{Adv}_\alpha(F)$ is a new abstract complexity measure that assigns cost α_i to querying the i^{th} input bit. We can then prove [HŠ05] that the quantity Adv_α composes in the following sense.

Theorem 8 (Composition Theorem [BS04, Amb03, LLS05, HŠ05]) *For any composite function $H = F \circ (G_1, \ldots, G_k)$, where $F : \{0,1\}^k \to \{0,1\}$ and $G_i : \{0,1\}^{N_i} \to \{0,1\}$ are partial boolean functions,*

$$\mathrm{Adv}_\alpha(H) = \mathrm{Adv}_\beta(F),$$

where $\beta_i = \mathrm{Adv}_{\alpha^i}(G_i)$, and $\alpha = (\alpha^1, \ldots, \alpha^k)$ is a k-tuple of strings $\alpha^i \in \Re^{+N_i}$.

A natural generalization of Grover's problem is the so-called k-fold search problem in which we are promised that exactly k entries of the input oracle x are one (so $|x| = k$), and the goal is to find all of these k indices. We say an algorithm A succeeds if it outputs a subset $S \subseteq [N]$ of size k and S contains all indices $i \in [N]$ for which $x_i = 1$. Thus, by definition, it fails even if it outputs all but one of the k indices. The k-fold search problem can be solved in $O(\sqrt{kn})$ queries, essentially by sequentially running Grover's search algorithm k times. Klauck, Špalek, and de Wolf [KŠW04] show that if the number of queries is less than $\epsilon\sqrt{kn}$ for some constant ϵ, then the success probability of A is exponentially small in k. They thus prove a strong direct product theorem for the k-fold search problem. One of the main elements of the proof is the polynomial method which we discuss in the next section.

In very recent work, Ambainis [Am05a] proposes an extension of the adversary method and uses it to reprove the strong direct product theorem of [KŠW04]. Though the following very brief description of the proof does not do full justice to the method, we hope it conveys some of the intuition on which [Am05a] is based. The algorithm runs on a uniform superposition of all inputs.

During the computation, the input register gets entangled with the workspace of the algorithm due to the queries to the oracle. We trace out the workspace and examine the eigenspaces of the density matrix of the input register. Due to symmetries, there are exactly $k+1$ eigenspaces, indexed by the number of ones the algorithm "knows" at that stage of the algorithm. In the beginning, all amplitude is in the 0^{th} eigenspace. One query can only move little amplitude from the i^{th} eigenspace to the $i+1^{\text{st}}$ eigenspace. If the algorithm has good success probability, the quantum amplitude in high eigenspaces must be significant, since the algorithm must "know" most of the k indices, which implies a lower bound on the query complexity.

7 Polynomial lower bounds

There are essentially two different methods known for proving lower bounds on quantum computations. The historically first method is the adversary method we discuss above. It was introduced in 1994 by Bennett, Bernstein, Brassard, and Vazirani, and published in 1997 in the SIAM Journal on Computing, in a special section that contains some of the most outstanding papers on quantum computing. The second method was introduced shortly after, in 1998, by Beals, Buhrman, Cleve, Mosca, and de Wolf [BB$^+$01], and implicitly used by Fortnow and Rogers in [FR99]. Their approach is algebraic and follows earlier very successful work on classical lower bounds via polynomials (see for instance Beigel's 1993 survey [Bei93] and Regan's 1997 survey [Reg97]). We first establish that any partial boolean function $F : S \to \{0,1\}$, where $S \subseteq \{0,1\}^N$, can be represented by a real-valued polynomial $p : \Re^N \to \Re$.

Definition 9 *Let $F : S \to \{0,1\}$ be a partial boolean function, where $S \subseteq \{0,1\}^N$. An N-variable polynomial p* represents *F if $p(x) = F(x)$ for all $x \in S$, and it* approximates *F if $|p(x) - F(x)| \leq \frac{1}{3}$ for all $x \in S$. The* degree *of F, denoted $\deg(F)$, is the minimal degree of a polynomial representing F. The* approximate degree *of F, denoted $\widetilde{\deg}(F)$, is the minimal degree of a polynomial approximating F.*

The crux in [BB$^+$01] is in showing that any quantum algorithm A computing some function F gives rise to some polynomial p_{A} that represents or approximates F.

Theorem 10 ([BB$^+$01]) *Let A be a quantum algorithm that computes a partial boolean function $F : S \to \{0,1\}$, where $S \subseteq \{0,1\}^N$, using at most T queries to the oracle O'_x. Then there exists an N-variate real-valued multilinear polynomial $p_{\mathsf{A}} : \Re^N \to \Re$ of degree at most $2T$, which equals the acceptance probability of A.*

Proof In this theorem, we use the oracle O'_x which is equivalent to the oracle O_x, since it allows for simple formulations. We first rewrite the action of O'_x as

$$\mathsf{O}'_x|i,b;z\rangle = (1-x_i)|i,b;z\rangle + x_i|i,b\oplus 1;z\rangle \tag{13}$$

where we define $x_i = 0$ for $i = 0$ so that we can simulate a non-query by querying x_i with $i = 0$. Suppose we apply O'_x on some superposition $\sum_{i,b,z} \alpha_{i,b,z}|i,b;z\rangle$ where each amplitude $\alpha_{i,b,z}$ is an

N-variate complex-valued polynomial in x of degree at most j. Then, by Eq. 13, the resulting state $\sum_{i,b,z} \beta_{i,b,z}|i,b;z\rangle$ is a superposition where each amplitude $\beta_{i,b,z}$ is an N-variate complex-valued polynomial in x of degree at most $j+1$. By proof by induction, after T queries, each amplitude can be expressed as a complex-valued polynomial in x of degree at most T. The probability that the final measurement yields the outcome 1, corresponding to accepting the input, is obtained by summing some of the absolute values of the amplitudes squared. The square of any of the absolute amplitudes can be expressed as a real-valued polynomial p_{A} in x of degree at most $2T$. Theorem 10 follows. □

The above theorem states that to any quantum algorithm A computing a boolean function $F : S \to \{0,1\}$, where $S \subseteq \{0,1\}^N$, we can associate an N-variate polynomial $p_{\mathsf{A}} : \Re^N \to \Re$ that expresses the acceptance probability of the algorithm on any given input. If algorithm A is exact, i.e., if A always stops and outputs the correct answer, then $p_{\mathsf{A}}(x) = F(x)$ for all $x \in S$, and thus p_{A} represents F. If A has bounded error, then $0 \leq p_{\mathsf{A}}(x) \leq 1/3$ if $F(x) = 0$ and $2/3 \leq p_{\mathsf{A}}(x) \leq 1$ if $F(x) = 1$, and thus p_{A} approximates F. The degree of p_{A} is at most twice the number of queries used by algorithm A. Consequently, the degree of a function is a lower bound on the quantum query complexity, up to a factor of two.

Corollary 11 (Polynomial method [BB$^+$01]) *For any partial boolean function $F : S \to \{0,1\}$, where $S \subseteq \{0,1\}^N$, we have $Q_E(F) \geq \deg(F)/2$ and $Q_2(F) \geq \widetilde{\deg}(F)/2$.*

8 Applying the polynomial method

The challenge in applying the polynomial method lies in the dimensionality of the input. Typically, the method is applied by first identifying a univariate or bivariate polynomial that captures essential properties of the problem, and then proving a lower bound on the degree of that polynomial. The second part is typically reasonably straightforward since polynomials have been studied for centuries and much is known about their degrees. The possibly simplest nontrivial example is when F is the threshold function Thr_t defined by $\mathsf{Thr}_t(x) = 1$ if and only if $|x| \geq t$. It is easy to see that $\deg(\mathsf{Thr}_t) = \Theta(N)$ for all nontrivial threshold functions, and thus $Q_E(\mathsf{Thr}_t) = \Omega(N)$. Paturi [Pat92] shows that $\widetilde{\deg}(\mathsf{Thr}_t) = \Theta\big(\sqrt{(t+1)(N-t+1)}\big)$, and we thus readily get that $Q_2(\mathsf{Thr}_t) = \Omega\big(\sqrt{(t+1)(N-t+1)}\big)$, which is tight by quantum counting [BH$^+$02, BB$^+$01]. This degree argument extends to any symmetric function F by writing F as a sum of threshold functions. The same tight lower bounds for symmetric functions can also be obtained by the unweighted adversary method (see the paragraph after Theorem 4).

For general non-symmetric functions, the polynomial method is, however, significantly harder to apply. For problems that are "close" to being symmetric, we can sometimes succeed in constructing a univariate or bivariate polynomial that yields a non-trivial lower bound. The first and, in our view, most important such a result was obtained by Aaronson in [Aar02] in which he proves a lower bound of $\Omega(N^{1/5})$ on any bounded-error quantum algorithm for the collision problem.

The collision problem is a non-boolean promise problem. The oracle is an N-tuple of positive integers between 1 and M, which we think of as a function $X : [N] \to [M]$. We model the oracle

O''_X so that a query to the i^{th} entry of the oracle returns the integer $X(i)$. Specifically, O''_X takes as input $|i, r; z\rangle$ and outputs $|i, r \oplus X(i); z\rangle$ where $0 \leq r < 2^m$ for $m = \lceil \log_2(M+1) \rceil$, and $r \oplus X(i)$ denotes bitwise addition modulo 2. We are promised that either X is a one-to-one function, or X is two-to-one, and the goal is to determine which is the case.

The result of Aaronson was shortly after improved by Shi [Shi02] to $\Omega(N^{1/4})$ for general functions $X : [N] \to [M]$, and to $\Omega(N^{1/3})$ in the case the range is larger than the domain by a constant factor, $M \geq \frac{3}{2}N$. The lower bounds of Aaronson and Shi appears as a joint article [AS04]. Finally, Kutin [Kut05] and Ambainis [Am05b] independently found remedies for the technical limitations in Shi's proof, yielding an $\Omega(N^{1/3})$ lower bound for all functions, which is tight by an algorithm that uses Grover search on subsets by Brassard, Høyer, and Tapp [BHM97].

The best lower bound for the collision problem that can be obtained using the adversary method is only a constant, since any one-to-one function is of large Hamming distance to any two-to-one function. Koiran, Nesme, and Portier [KNP05] use the polynomial method to prove a lower bound of $\Omega(\log N)$ for Simon's problem [Sim97], which is tight [Sim97, BH97]. Simon's problem is a partial boolean function having properties related to finite abelian groups. Also for this problem, the best lower bound that can be obtained using the adversary method is a constant.

In contrast, for any *total* boolean function $F : \{0,1\}^N \to \{0,1\}$, the adversary and polynomial method are both polynomially related to block sensitivity,

$$\sqrt{\mathrm{bs}(F)/6} \leq \widetilde{\deg}(F) \leq \deg(F) \leq \mathrm{bs}^3(F) \tag{14}$$

$$\sqrt{\mathrm{bs}(F)} \leq \mathrm{Adv}(F) \leq \mathrm{bs}^2(F). \tag{15}$$

It follows from [BB$^+$01] that $\deg(F) \leq \mathrm{bs}^3(F)$, and from Nisan and Szegedy [NS92] that $6\widetilde{\deg}(F)^2 \geq \mathrm{bs}(F)$. Buhrman and de Wolf [BW02] provide an excellent survey of these and other complexity measures of boolean functions.

The polynomial lower bound is known to be inferior to the weighted adversary method for some total boolean functions. In [Amb03], Ambainis gives a boolean function $F : \{0,1\}^4 \to \{0,1\}$ on four bits, which can be described as "the four input bits are sorted" [LLS05], for which $\deg(F) = 2$ and for which there exists an adversary matrix Γ^F satisfying that $\lambda(\Gamma^F)/\max_i \lambda(\Gamma^F_i) = 2.5$. We compose the function with itself and obtain a boolean function $F_2 = F \circ (F, F, F, F) : \{0,1\}^{16} \to \{0,1\}$ defined on 16 bits for which $\deg(F_2) = 4$, and for which $\lambda(\Gamma^{F_2})/\max_i \lambda(\Gamma^{F_2}_i) = 2.5^2$, by the composition theorem. Iterating n times, yields a function F on $N = 4^n$ bits of degree $\deg(F) = 2^n$, with spectral lower bound $2.5^n = \deg(F)^{1.32\cdots}$, by the composition theorem. The thus constructed function F is an example of an iterated function of low degree and high quantum query complexity. It is the currently biggest known gap between the polynomial method and the adversary method for a total function. Another iterated total function for which the adversary methods yield a lower bound better than the degree, is the function described by "all three input bits are equal" [Amb03].

The polynomial method is very suitable when considering quantum algorithms computing functions with error ϵ that is sub-constant, whereas the adversary method is not formulated so as to capture such a fine-grained analysis. Buhrman, Cleve, de Wolf, and Zalka [BC$^+$99] show that any quantum algorithm for Grover's problem that succeeds in finding an index i for which $x_i = 1$ with probability at least $1 - \epsilon$, provided one exists, requires $\Omega(\sqrt{N \log(1/\epsilon)})$ queries to the oracle,

which is tight. A possibly more familiar example is that any polynomial approximating the parity function with any positive bias $\epsilon > 0$ (as opposed to bias $\frac{1}{6}$ where $\frac{1}{6} = \frac{2}{3} - \frac{1}{2}$) has degree N, since any such polynomial gives rise to a univariate polynomial of no larger degree with N roots. Hence, any quantum algorithm computing the parity function with arbitrary small bias $\epsilon > 0$ requires $N/2$ queries to the oracle, which is tight.

A useful property of representing polynomials is that they compose. If p is a polynomial representing a function F, and polynomials $q_1, q_2, \ldots, q_k$ represent functions $G_1, \ldots, G_k$, then $p \circ (q_1, \ldots, q_k)$ represents $F \circ (G_1, \ldots, G_k)$, when well-defined. This composition property does not hold for approximating polynomials: if each sub-polynomial q_i takes the value 0.8, say, then we cannot say much about the value $p(0.8, \ldots, 0.8)$ since the value of p on non-integral inputs is not restricted by the definition of being an approximating polynomial. To achieve composition properties, we require that the polynomials are insensitive to small variations of the input bits. Buhrman, Newman, Röhrig, and de Wolf give in [BN$^+$05] a definition of such polynomials, and refer to them as being robust.

Definition 12 (Robust polynomials [BN$^+$05]) *An approximate N-variate polynomial p is* robust *on $S \subseteq \{0,1\}^N$ if $|p(y) - p(x)| \leq \frac{1}{3}$ for every $x \in S$ and $y \in \Re^M$ such that $|y_i - x_i| \leq \frac{1}{3}$ for every $i = 1, \ldots, M$. The* robust degree *of a boolean function $F : S \to \{0,1\}$, denoted $\mathrm{rdeg}(F)$, is the minimal degree of a robust polynomial approximating F.*

Robust polynomials compose by definition. Buhrman et al. [BN$^+$05] show that the robust degree of any total function $F : \{0,1\}^N \to \{0,1\}$ is $O(N)$ by giving a classical algorithm that uses a quantum subroutine for Grover's problem [Gro96] which is tolerant to errors, due to Høyer, Mosca, and de Wolf [HMW03]. Buhrman et al. [BN$^+$05] also show that $\mathrm{rdeg}(F) \in O(\widetilde{\deg}(F) \log \widetilde{\deg}(F))$ by giving a construction for turning any approximating polynomial into a robust polynomial at the cost of at most a logarithmic factor in the degree of F. This implies that for any composite function $H = F \circ (G, \ldots, G)$, we have $\widetilde{\deg}(H) \in O(\widetilde{\deg}(F)\widetilde{\deg}(G) \log \widetilde{\deg}(F))$. It is not known whether this is tight. Neither is it known if the approximate degree of H can be significantly smaller than the product of the approximate degrees of F and G. The only known lower bound on the approximate degree of H is the trivial bound $\Omega(\widetilde{\deg}(F) + \widetilde{\deg}(G))$.

An and-or tree of depth two is a composed function $F \circ (G, \ldots, G)$ in which the outer function F is the logical AND of $\sqrt{N}$ bits, and the inner function G is the logical OR of $\sqrt{N}$ bits. By the unweighted adversary method, computing and-or trees of depth two requires $\Omega(\sqrt{N})$ queries. Høyer, Mosca, and de Wolf [HMW03] give a bounded-error quantum algorithm that uses $O(\sqrt{N})$ queries, which thus is tight. The existence of that algorithm implies that there exists an approximating polynomial for and-or tree of depth two of degree $O(\sqrt{N})$. No other characterization of an approximating polynomial for and-or trees of depth two of degree $O(\sqrt{N})$ is currently known. The best known lower bound on the approximate degree of and-or trees of depth two is $\Omega(N^{1/3})$, up to logarithmic factors in N, by a folklore reduction from the element distinctness problem on $\sqrt{N}$ integers [AS04].

9 Concluding remarks

We have been focusing on two methods for proving lower bounds on quantum query complexity: the adversary method and the polynomial method. Adversary lower bounds are in general easy to compute, but are limited by the certificate complexity. Known lower bounds are constructed by identifying hard input pairs, finding weights accordingly, and computing either the spectral norm of some matrices, or applying the weighted method. Polynomial lower bounds may yield stronger bounds, but are hard to prove. Known lower bounds by the polynomial methods are constructed by identifying symmetries within the problem, reducing the number of input variables to one or two, and proving a lower bound on the degree of the reduced polynomial.

Barnum, Saks, and Szegedy give in [BSS03] a third lower bound method that exactly characterizes the quantum query complexity, but this strength turns out also to be its weakness: it is very hard to apply and every known lower bound obtained by the method can also be shown by one of the other two methods. In a very recent work, Ambainis [Am05a] extends the adversary method and uses it to reprove a strong direct product theorem by Klauck, Špalek, and de Wolf [KŠW04] obtained by techniques that include the polynomial method. Klauck et al. [KŠW04] show that their strong direct product theorem implies good quantum time-space tradeoffs, including a quantum lower bound of $T^2 \cdot S = \Omega(N^3)$ for sorting. A significant body of work have been conducted on lower bounds on communication complexity. We refer to de Wolf's excellent survey [Wol02] as a possible starting point.

There is a range of problems for which we do not currently know tight quantum query bounds. One important example is binary and-or trees of logarithmic depth. A binary and-or tree on $N = 4^n$ variables is obtained by iterating the function $F(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$ in total n times. The classical query complexity for probabilistic algorithms is $\Theta(N^{0.753})$ [SW86, Sni85, San95]. No better bounded-error quantum algorithm is known. The best known lower bound on the quantum query complexity is $\Omega(\sqrt{N})$ by embedding the parity function on $\sqrt{N}$ bits and noting that the parity function has linear query complexity, which can be shown by either method.

Magniez, Santha, and Szegedy give in [MSS05] a quantum algorithm for determining if a graph on N vertices contains a triangle which uses $O(N^{1.3})$ queries to the adjacency matrix. The best known lower bound is $\Omega(N)$ by the unweighted adversary method, and has been conjectured not to be tight [Amb03]. The problem of triangle-identification is an example of a graph property, which is a set of graphs closed under isomorphism. Sun, Yao, and Zhang [SYZ04] show that there exists a non-trivial graph property of quantum query complexity $O(\sqrt{N})$, up to logarithmic factors in N.

Gasarch, in a survey on private information retrieval, published in this Computational Complexity Column in the Bulletin [Gas04], writes: "A field is interesting if it answers a fundamental question, or connects to other fields that are interesting, or uses techniques of interest." It is our hope that the reader will find that the surveyed area of quantum lower bounds fulfills each of those three criteria.

Acknowledgments

We thank Michal Koucký and Kolja Vereshchagin for discussions on the proof of the spectral adversary bound.

References

[Aar02] S. Aaronson. Quantum lower bound for the collision problem. In *Proceedings of 34th ACM Symposium on Theory of Computing*, pages 635–642, 2002.

[Aar04] S. Aaronson. Lower bounds for local search by quantum arguments. In *Proceedings of 36th ACM Symposium on Theory of Computing*, pages 465–474, 2004.

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64:750–767, 2002.

[Amb03] A. Ambainis. Polynomial degree vs. quantum query complexity. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*, pages 230–239, 2003.

[Am05a] A. Ambainis. *A new quantum lower bound method, with an application to strong direct product theorem for quantum search*. quant-ph/0508200, 2005.

[Am05b] A. Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1:37–46, 2005.

[AS04] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

[BB+97] H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[BB+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001.

[BC+99] H. Buhrman, R. Cleve, R. de Wolf, and Ch. Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 358–368, 1999.

[Bei93] R. Beigel. The polynomial method in circuit complexity. In *Proceedings of the 8th Annual Structure in Complexity Theory Conference*, IEEE Computer Society Press. pages 82–95, 1993.

[BH97] G. Brassard and P. Høyer. An exact quantum polynomial-time algorithm for Simon's problem. In *Proceedings of Fifth Israeli Symposium on Theory of Computing and Systems*, pages 12–23, 1997.

[BHM97] G. Brassard, P. Høyer, and A. Tapp. Quantum algorithm for the collision problem. *SIGACT News*, 28:14–19, 1997.

[BH+02] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, AMS Contemporary Mathematics Series, Volume 305, 2002.

[BN+05] H. Buhrman, I. Newman, H. Röhrig, and R. de Wolf. Robust quantum algorithms and polynomials. In *Proceedings of 22nd International Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 3404, 593–604, 2005.

[BS04] H. Barnum and M. Saks. A lower bound on the quantum query complexity of read-once functions. *Journal of Computer and Systems Sciences*, 69(2):244–258, 2004.

[BSS03] H. Barnum, M. Saks, and M. Szegedy. Quantum query complexity and semi-definite programming. In *Proceedings of the 18th IEEE Conference on Computational Complexity*, pages 179–193, 2003.

[BW92] C. H. Bennett and S. J. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Physical Review Letters*, 69(20):2881–2884, 1992.

[BW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[CD+98] R. Cleve, W. van Dam, M. Nielsen, and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Proceedings of the 1st NASA QCQC conference*, Lecture Notes in Computer Science 1509, pages 61–74, 1998.

[DH+04] Ch. Dürr, M. Heiligman, P. Høyer, and M. Mhalla. Quantum query complexity of some graph problems. In *Proceedings of 31st International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 3142, pages 481–493, 2004.

[DJ92] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society, London*, A439:553–558, 1992.

[EHK04] M. Ettinger and P. Høyer, and E. Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters*, 91(1):43-48, 2004.

[FGGS] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. *Invariant quantum algorithms for insertion into an ordered list*. quant-ph/9901059, 1999.

[FR99] L. Fortnow and J. D. Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59(2):240–252, 1999.

[Gas04] W. Gasarch. A survey on private information retrieval. The computational complexity column, in the *Bulletin of the EATCS*, 82:72–107, 2004.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[Gro05] D. Gross. *The Future of Physics.* CERN Colloquium, January 26, 2005. CERN, Switzerland.

[HMW03] P. Høyer, M. Mosca, and R. de Wolf. Quantum search on bounded-error inputs. In *Proceedings of 30th International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 2719, pages 291–299, 2003.

[HNS02] P. Høyer, J. Neerbek, and Y. Shi. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4):429–448, 2002.

[Hol73] A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973. English translation is *Problems in Information Transmission*, 9:177–183, 1973.

[HŠ05] P. Høyer and R. Špalek. *Tight adversary bounds for composite functions*. quant-ph/0509067, 2005.

[JLB05] M. B. Jacokes, A. J. Landahl, and E. Brookes. *An improved quantum algorithm for searching an ordered list.* Manuscript, 2005.

[KNP05] P. Koiran, V. Nesme and N. Portier. A quantum lower bound for the query complexity of Simon's problem. In *Proceedings of 32nd International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 3580, pages 1287–1298, 2005.

[KŠW04] H. Klauck, R. Špalek, and R. de Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pages 12–21, 2004.

[Kut05] S. Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1:29–36, 2005.

[LLS05] S. Laplante, T. Lee, and M. Szegedy. The quantum adversary method and formula size lower bounds. In *Proceedings of 20th IEEE Conference on Computational Complexity*, 2005.

[LM04] S. Laplante and F. Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. In *Proceedings of 19th IEEE Conference on Computational Complexity*, pages 294–304, 2004.

[Mat90] R. Mathias. The spectral norm of a nonnegative matrix. *Linear Algebra and its Applications*, 139:269–284, 1990.

[MSS05] F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. In *Proceedings of 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1109–1117, 2005.

[NS92] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. In *Proceedings of 24th ACM Symposium on Theory of Computing*, pages 462–467, 1992.

[NW99] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of 31st ACM Symposium on Theory of Computing*, pages 384–393, 1999.

[Pat92] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In *Proceedings of the 24th ACM Symposium on Theory of Computing*, pages 468–474, 1992.

[Reg97] K. Regan. Polynomials and combinatorial definitions of languages. In *Complexity Theory Retrospective II*, Springer-Verlag, pages 261–293, 1997.

[San95] M. Santha. On the Monte Carlo decision tree complexity of read-once formulae. *Random Structures and Algorithms*, 6(1):75–87, 1995.

[Sei05] Ch. Seife. "What are the limits of conventional computing?" *Science*, 309(5731):96, 1 July 2005.

[Shi02] Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. In *Proceedings of the 43rd Annual Symposium on the Foundations of Computer Science*, pp. 513-519, 2002.

[Sim97] D. R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[Sni85] M. Snir. Lower bounds on probabilistic decision trees. *Theoretical Computer Science*, 38:69–82, 1985.

[SS04] M. Santha and M. Szegedy. Quantum and classical query complexities of local search are polynomially related. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 494–501, 2004.

[ŠS05] R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. In *Proceedings of 32nd International Colloquium on Automata, Languages and Programming* , Lecture Notes in Computer Science 3580, pages 1299–1311, 2005.

[SW86] M. Saks and A. Wigderson. Probabilistic boolean decision trees and the complexity of evaluating games trees. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 29–38, 1986.

[SYZ04] X. Sun, A. C. Yao, and S. Zhang. Graph properties and circular functions: How low can quantum query complexity go? In *Proceedings of 19th IEEE Conference on Computational Complexity*, pages 286–293, 2004.

[Val03] L. G. Valiant. Three problems in computer science. *Journal of Association of Computing Machinery*, 50(1):96–99, 2003.

[Wol02] R. de Wolf. Quantum communication and complexity. *Theoretical Computer Science*, 287(1): 337–353, 2002.

[Zal99] Ch. Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, 1999.

[Zha04] S. Zhang. On the power of Ambainis's lower bounds. In *Proceedings of 31st International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 3142, pages 1238–1250, 2004.

[Zha05] S. Zhang. *(Almost) tight bounds for randomized and quantum local search on hypercubes and grids*. quant-ph/0504085, 2005.