PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

On the power of quantum computation

Umesh Vazirani

Phil. Trans. R. Soc. Lond. A 1998 **356**, 1759-1768
doi: 10.1098/rsta.1998.0247

The Royal Society

On the power of quantum computation

By Umesh Vazirani

Computer Science Division, University of California, Berkeley, CA 94720, USA

This paper surveys the use of the 'hybrid argument' to prove that quantum corrections are insensitive to small perturbations. This property of quantum computations is used to establish that quantum circuits are robust against inaccuracy in the implementation of its elementary gates. The insensitivity to small perturbations is also used to establish lower-bounds, including showing that relative to an oracle, the class NP requires exponential time on a quantum computer; and that quantum algorithms are polynomially related to deterministic algorithms in the black-box model.

Keywords: quantum computation; NP; oracle; hybrid argument

1. Introduction

Quantum computation is an exciting area that lies at the foundations of both quantum physics and computer science. Quantum computers appear to violate the modern form of the Church–Turing thesis, which states that any 'reasonable' model of computation can be simulated by a probabilistic Turing machine with at most polynomial factor simulation overhead. The first evidence that quantum computers violate this thesis was given by Bernstein & Vazirani (1993, 1997), building upon earlier work of Deutsch & Jozsa (1992). This was followed by an important group-theoretic quantum algorithm by Simon (1994, 1997), and then by the remarkable result of Shor (1994, 1997) showing that factoring and computing discrete logs are computable in polynomial time on a quantum computer. In view of these results, it is natural to ask whether quantum computers can solve all problems in the class NP (or $\mathrm{NP} \cap \text{co-NP}$) in polynomial time. Bennett *et al.* (1997) gave evidence that this question is unlikely to be resolved without a major breakthrough in complexity theory, by showing that relative to a random oracle, $\mathrm{NP} \not\subseteq \mathrm{BQTIME}(o(2^{n/2}))$. This result is the best possible, up to constant factors, since there is a matching upper bound that follows from Grover's (1996) quantum search algorithm. Boyer *et al.* (1996) obtain the exact constants in the upper and lower bounds, thus exhibiting a tight bound on quantum search. Bennett *et al.* (1997) also showed that relative to a random permutation oracle $\mathrm{NP} \cap \text{co-NP} \not\subseteq \mathrm{BQTIME}(o(2^{n/3}))$.

As in all oracle lower bounds, the results of Bennett *et al.* (1997) are proved by establishing a lower bound on the number of oracle queries that the algorithm must make. More generally, in the black-box model, the algorithm is not given the input explicitly, but is instead allowed to query bits of the input. The cost of the algorithm on a given input is defined to be the number of queries to the input, independent of the time spent by the algorithm between queries. Let $D(f)$ denote the number of queries made by a deterministic algorithm to compute a Boolean function $f : \{0,1\}^n \to \{0,1\}$, and let $Q(f)$ denote the number of queries made by a quantum algorithm that computes f with error probability less by $\frac{1}{3}$. Recently, Beals *et al.* (1998) introduced a new framework, involving polynomials, for establishing lower bounds in the black-box model; they proved a general result showing that $D(f) = O(Q(f)^6)$. In this paper we prove the weaker bound $D(f) = O(Q(f)^8)$, and sketch an outline of the $D(f) = O(Q(f)^6)$ result.

Finally we consider the question: how 'reasonable' is the quantum computation model? There are several 'unreasonable' classical models of computation, in which it is possible to factor numbers or even solve NP-complete problems in polynomial time (see Shamir (1979) for a particularly delightful example). In each case, these models rely on being able to carry out operations on numbers with exponentially many bits in a single computational step. This is unrealistic, because in any implementation, such a number must be represented as a physical quantity which we can manipulate with only limited precision. Quantum circuits or quantum Turing machines are designed to be discrete models of computation, except for the fact that in each case we must be able to carry out a rotation through angle θ of a single qubit. Does this introduction of a real number θ into the model make it 'unreasonable'; i.e. to what accuracy must we design our physical apparatus to carry out the 'rotation through angle θ'? Bernstein & Vazirani (1993, 1997) showed that in any quantum circuit with m gates, if each rotation gate is accurate to within $\delta\theta \leqslant \epsilon/4m$ then the output of the approximate circuit is indistinguishable from the output of the ideal circuit except with probability ϵ. We give a proof of this theorem. In fact, we use the same proof technique for all the results in this paper. The technique is known as the 'hybrid argument' among cryptographers. Using quantum error-correction techniques, fault-tolerance quantum circuits can be created that are resilient to constant error in the rotation gates of the circuit, independent of the size of the circuit (Aharanov & Ben-Or 1996; Gottesman 1997).

How does one explain the power of quantum computation? The dimension of the Hilbert space associated with an n-qubit system is 2^n. Therefore, just describing the state of this system requires 2^n complex numbers. Moreover, nature must update the 2^n complex numbers to evolve the system in time. This is an extravagant amount of work for nature to perform, even for systems consisting of as few as 200 qubits, since 2^{200} is larger than estimates for the number of particles in the universe. Although this is part of an explanation, we can get further insights by considering our original question more closely. We first observe that, even in classical physics, nature performs computations which are capable of solving problems such as factoring and satisfiability. The difficulty lies in our ability to harness this for useful computation. For example, in classical physics, the state of a system with n degrees of freedom is described by n real numbers. Moreover, nature updates these real numbers by performing elementary operations, such as addition, upon them. As pointed out above, models such as this, with infinite-precision (or even exponential-precision) arithmetic are capable of carrying out tasks such as factoring or satisfiability in polynomial time (Shamir 1979; Vergis *et al.* 1986). What distinguishes quantum computation from classical computation is our ability to prepare the system to solve a computational problem of our choice. If we try to build a classical device to carry out a desired computation, the imprecision in the realization of the system effectively leaves us with only a few bits of information per degree of freedom of the system. Therefore we are effectively restricted to a cellular automaton or Turing machine. In the case of quantum computation, our access to the computation performed by nature is more subtle. When we perform an approximate rotation gate, the amplitudes in the resulting superposition are inaccurate; but in a very correlated manner. The approximate transformation is still unitary: it is an approximation in that it is close to the ideal transformation in operator norm. Theorem 2.2 shows that a moderate degree of approximation in operator norm is sufficient to guarantee that the results of the approximate circuit are close to the ideal circuit. Now imagine for a moment that in the quantum mechanical case our access to the computation performed by nature had been more coarse. Suppose that each time we carried out a quantum gate, we could only specify the amplitudes in the resulting superposition accurate to within $1/k$ if we used effort proportional to k. It is not hard to show that in this case, too, we are effectively restricted to carrying out computations that can be simulated in polynomial time on a classical Turing machine. The moral is that the power of quantum computation should be ascribed not only to the exponential parallelism in the quantum system, but also to the fact that we can harness that computation despite our noisy and inaccurate access to the system.

2. Quantum circuits with approximate gates

Consider a quantum circuit that operates on n qubits and consists of a sequence of m elementary quantum gates. Then this circuit applies a sequence of m unitary transformations $U_1, \ldots, U_m$ to some initial state vector $|\phi_0\rangle \in \mathcal{C}^{2^n}$. How sensitive is the output of the circuit to perturbations in the transformations carried out by the gates? For example, consider an elementary gate that performs a rotation through θ on a single quantum bit. The unitary transformation on the single bit is given by

$$\begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix},$$

and therefore the unitary transformation on $\mathcal{C}^{2^n}$ is given by

$$\begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} \otimes I,$$

where I is the identity matrix on $\mathcal{C}^{2^{n-1}}$. In any implementation of this rotation gate, we have to allow for some imprecision in achieving the desired angle θ, and must assume that the actual gate that is implemented achieves some rotation θ' where $|\theta' - \theta| \leqslant \delta\theta$. The corresponding unitary transformation U' satisfies $\|U - U'\| \leqslant \delta\theta$†. How severely is the output of the circuit changed if each U_k is replaced by a perturbation V_k such that $\|V_k - U_k\| \leqslant \delta$?

Denote by $|\phi_k\rangle$ the state of the n qubits after the application of the first k gates. Let us start by considering the change in the final state vector $|\phi_m\rangle$ in the special case that only one of the gates, say the kth one, is perturbed. In this case, the state of the n qubits after the perturbed kth gate is $|\phi'_k\rangle = V_k|\phi_{k-1}\rangle$ instead of $|\phi_k\rangle = U_k|\phi_{k-1}\rangle$. Therefore the error after the kth step is $|\phi'_k\rangle - |\phi_k\rangle = (V_k - U_k)|\phi_{k-1}\rangle$. How do these two state vectors (and therefore the error) evolve under the action of the remaining gates in the circuit. The key point is that unitary evolution preserves dot product, and therefore the norm of the error vector is preserved by the subsequent gates in

† The norm of a linear operator A is

$$\|A\| \stackrel{\text{def}}{=} \max_{v:\|v\|=1} \|Av\|.$$

the circuit. Therefore the norm of the final error vector is $\|(V_k - U_k)|\phi_{k-1}\rangle\| \leqslant \|V_k - U_k\| \leqslant \delta$.

The formal content of the previous paragraph can be expressed in the following calculation. Let $|\psi\rangle = U_m \ldots U_1|\phi_0\rangle$ and let $|\psi'\rangle = U_m \ldots U_{k+1}V_kU_{k-1} \ldots U_1|\phi_0\rangle$. Then

$$\begin{aligned}\||\psi'\rangle - |\psi\rangle\| &= \|U_m \ldots U_{k+1}U_kU_{k-1} \ldots U_1|\phi_0\rangle - U_m \ldots U_{k+1}V_kU_{k-1} \ldots U_1|\phi_0\rangle\| \\ &= \|U_m \ldots U_{k+1}(U_k - V_k)U_{k-1} \ldots U_1|\phi_0\rangle\| \\ &= \|(U_k - V_k)|\phi_{k-1}\rangle\| \\ &\leqslant \|V_k - U_k\|.\end{aligned}$$

Notation. Denote by $\mathcal{D}(\psi)$ the probability distribution that results from a measurement of $|\psi\rangle$ in the computational basis.

The following lemma from Bernstein & Vazirani (1993, 1997) shows that a bound on the norm of the final error vector can be directly translated into an upper bound on the distance between the corresponding output distributions. Here the distance between two distributions is the 1-norm or the total variation distance.

Lemma 2.1. *If $\||\psi'\rangle - |\psi\rangle\| \leqslant \epsilon$, then $\|\mathcal{D}(\psi') - \mathcal{D}(\psi)\|_1 \leqslant 4\epsilon$.*

Now consider the general case where each gate U_k is replaced by a perturbation V_k such that $\|U_k - V_k\| \leqslant \delta$.

We show that the contributions to the final error vector from these contributions accumulate additively by a technique that cryptographers will recognize as a 'hybrid argument'. Consider a sequence of runs of the circuit; in the jth run, the last j gates are accurate, but the first $m - j$ gates are replaced by their perturbations (see figure 1). Denote by $|\psi_j\rangle$ the final state vector on the jth run. Clearly, $|\psi_m\rangle$ is the final state vector of the ideal circuit, and $|\psi_0\rangle$ is the final state vector of the perturbed circuit. The main point of this construction is that successive runs differ in only one gate, and therefore we can bound the difference in their final state vectors as we did above by the norm of the difference between the single perturbed transformation and its accurate counterpart, and therefore by δ. Since $|\psi_0\rangle - |\psi_m\rangle$ can be expressed as a sum of m such successive differences, we can use the triangle inequality to derive the bound $\||\psi_0\rangle - |\psi_m\rangle\| \leqslant m\delta$. Finally, appealing to lemma 2.1, if $\delta \leqslant \epsilon/(4m)$, then the output distribution of the ideal and the perturbed circuit differ by at most ϵ.

Theorem 2.2. *If the gates of a quantum circuit $U_1, \ldots, U_m$ are replaced by perturbations $V_1, \ldots, V_m$ such that $\|U_k - V_k\| \leqslant \epsilon/(4m)$, then the output distribution of the perturbed circuit is within ϵ (in 1-norm) of the output distribution of the ideal circuit.*

3. A lower bound for quantum search

In the oracle model, the quantum algorithm is given access to an oracle $A : \{0,1\}^* \to \{0,1\}$. Think of A as a subroutine that the quantum algorithm can invoke, but one whose underlying program it is not allowed to see. For simplicity, we assume that each oracle access counts as one step of computation. Consider the following problem: on input 1^n (n in unary notation), is there an x of length n such that $A(x) = 1$? This problem is clearly in the class NP relative to the oracle A,

step								
1	V_1	V_1	V_1	V_1		V_1	V_1	U_1
2	V_2	V_2	V_2	V_2	$\dots$	V_2	U_2	U_2
3	V_3	V_3	V_3	V_3		U_3	U_3	U_3
$\vdots$		$\vdots$			$\dots$		$\vdots$	
$m-2$	V_{m-3}	V_{m-3}	V_{m-3}	V_{m-2}		U_{m-2}	U_{m-2}	U_{m-2}
$m-2$	V_{m-2}	V_{m-2}	V_{m-2}	U_{m-2}		U_{m-2}	U_{m-2}	U_{m-2}
$m-1$	V_{m-1}	V_{m-1}	U_{m-1}	U_{m-1}	$\dots$	U_{m-1}	U_{m-1}	U_{m-1}
m	V_m	U_m	U_m	U_m		U_m	U_m	U_m
result	$\vert\psi_0\rangle$	$\vert\psi_1\rangle$	$\vert\psi_2\rangle$	$\vert\psi_3\rangle$	$\dots$	$\vert\psi_{m-2}\rangle$	$\vert\psi_{m-1}\rangle$	$\vert\psi_m\rangle$

Figure 1. The hybrid argument.

since a yes answer can be verified by a single probe into A (together with a trivial check). In this section, we show that this problem cannot be solved in asymptotically less than $2^{n/2}$ steps by any quantum algorithm with bounded error. Thus $\mathrm{NP}^A \not\subseteq \mathrm{BQTIME}(o(2^{n/2}))^A$. The actual results of Bennett *et al.* (1997) on which this section is based are somewhat stronger. They show that that relative to a random oracle, $\mathrm{NP} \not\subseteq \mathrm{BQTIME}(o(2^{n/2}))$. They also show that relative to a random permutation oracle $\mathrm{NP} \cap \text{co-NP} \not\subseteq \mathrm{BQTIME}(o(2^{n/3}))$.

How significant are these oracle results in view of the recent non-relativizing results in complexity theory, such as IP = PSPACE (Shamir 1990), and $\mathrm{NP} = \mathrm{PCP}(\log n, 1)$ (Arora *et al.* 1992). Arora *et al.* (1994) formalize the folk notion that any result in complexity theory that is proved using only simulation and diagonalization arguments holds relative to all oracles. However, they also pointed out that there is one non-relativizing technique in complexity theory, namely the Cook–Levin theorem (Cook 1971; Levin 1973). The non-relativizing form of the Cook–Levin theorem says that any language with a polynomial-time checkable proof of membership also has a log space checkable proof of membership. Arora *et al.* (1994) also argue that all the recent non-relativizing results in complexity theory depend upon the (prominent) use of the Cook–Levin theorem in their proof. In view of this, we can conclude that the results in Bennett *et al.* (1997) indicate that there are two ways to design a polynomial-time quantum algorithm for NP (or $\mathrm{NP} \cap \text{co-NP}$). (a) By using the Cook–Levin theorem in an essential way in the design of the algorithm. This appears quite challenging since it requires some relationship between the computational resources required for checking a proof of membership and quantum algorithms. (b) By designing a new non-relativizing technique. Complexity theorists have been searching for such techniques for the last two decades with no success.

Lower bounds in the oracle model are invariably proved by establishing the desired bound on the number of queries to the oracle by the algorithm, independent of the actual amount of computation performed by the algorithm between queries. This motivates the following model. In the quantum analogue of the black-box or Boolean decision tree model, the only cost associated with a quantum algorithm is the number of accesses to the input. All other computation is free. Thus if the input to the algorithm is $x \in \{0,1\}^n$, then a query of the form $\sum_{j=1}^{n} \alpha_j |j\rangle$ results in the state $\sum_{j=1}^{n} \alpha_j |j\rangle |x_j\rangle$. In this section we show a lower bound of $\Omega(\sqrt{n})$ in the black-box model for the problem of deciding whether or not the input string x is equal to 0^n. This implies a similar bound for the search problem: find j such that $x_j = 1$. This is tight up to constant factors because Grover's algorithm shows how to perform quantum search in $O(\sqrt{n})$ steps. A tight bound on the constants was given by Boyer *et al.* (1996).

Sensitivity to perturbation. Fix a quantum algorithm $\mathcal{A}$. All the definitions and discussion that follow are with respect to this fixed algorithm $\mathcal{A}$. Recall that the state of $\mathcal{A}$ between successive queries can be written as $|\phi\rangle = \sum_c \alpha_c|c\rangle$, where c ranges over all computational basis states (the possible classical states of the memory). Now, during the query, each such computational basis state c probes a particular bit position. The following definitions are motivated by the question: how sensitive is the output of $\mathcal{A}$ to the modification of the input in a few bit positions?

Definition. The *query magnitude* at bit position j of $|\phi\rangle = \sum_c \alpha_c|c\rangle$ is defined to be $q_j(|\phi\rangle) = \sum_{c\in C_j}|\alpha_c|^2$, where C_j is the set of all computational basis states c that query bit position j.

Assume that $\mathcal{A}$ runs for m steps on inputs of length n. Fix an input $x \in \{0,1\}^n$. The run of $\mathcal{A}$ on input x can be described by a sequence of states $|\phi_0\rangle, \ldots, |\phi_m\rangle$, where $|\phi_k\rangle$ is the state of $\mathcal{A}$ just before the $k+1^{st}$ query.

Definition. The query magnitude at bit position j of $\mathcal{A}$ on input x is defined to be $q_j(x) = \sum_{k=0}^{m-1} q_j(|\phi_k\rangle)$.

It is tempting to think of $q_j(x)$ as the probability that $\mathcal{A}$ probes bit position j and therefore conclude that it provides an upper bound on the probability that the output of $\mathcal{A}$ changes if x_j is flipped. Of course this is nonsense since there are no probabilities during the execution of the algorithm, just probability amplitudes that might interfere constructively or destructively. Nevertheless, we will show that if the total query magnitude of bit position j is very small, then $\mathcal{A}$ cannot distinguish whether the input x is modified by flipping its jth bit.

Given two strings $x, y \in \{0,1\}^n$, denote by $\Delta(x,y)$ the set of bit positions at which x and y differ. i.e. $\Delta(x,y) = \{j : x_j \neq y_j\}$.

Lemma 3.1 (Swapping lemma). *Let $|\phi_x\rangle$ and $|\phi_y\rangle$ denote the final states of $\mathcal{A}$ on inputs x and y, respectively. Then*

$$\||\phi_x\rangle - |\phi_y\rangle\| \leqslant \sqrt{m \sum_{j\in\Delta(x,y)} q_j(x)}.$$

Proof. The proof is by a hybrid argument. We compare a run of $\mathcal{A}$ on input x with a run on input y by defining a sequence of hybrid runs. In the kth hybrid run, the first k queries of $\mathcal{A}$ are answered according to input x, and the remaining $m-k$ queries are answered according to input to y. Denote by $|\phi_{k,t}\rangle$ the state of the kth run of $\mathcal{A}$ just before the $t+1$st query. Of course $|\phi_{k,0}\rangle = |\phi_0\rangle$ for all runs k, and $|\phi_{m,m}\rangle = |\phi_x\rangle$ and $|\phi_{0,m}\rangle = |\phi_y\rangle$.

Let us compare the final states of two successive hybrid runs. The final state of the kth run is $|\phi_{k,m}\rangle$, and the final state of the $k+1^{st}$ run is $|\phi_{k+1,m}\rangle$. Clearly the first k steps of the kth and $k+1$st runs are identical, since in each case the queries are answered according to the input x. Therefore $|\phi_{k,k}\rangle = |\phi_{k+1,k}\rangle = |\phi_k\rangle$. Now the k+1st query on the kth hybrid run is answered according to input x, whereas on the $k+1$st hybrid run it is answered according to input y. Since in each case the state of $\mathcal{A}$ while making the query is $|\phi_k\rangle$, and since x and y differ only on bit positions $j \in \varDelta(x,y)$, it follows that $\| |\phi_{k,k+1}\rangle - |\phi_{k+1,k+1}\rangle \|^2 \leqslant \sum_{j\in\varDelta(x,y)} q_j(|\phi_k\rangle)$. The remaining queries are answered according to input y in both hybrid runs. Therefore, since unitary evolution preserves dot product, we have that $\| |\phi_{k,k+1}\rangle - |\phi_{k+1,k+1}\rangle \| = \| |\phi_{k,m}\rangle - |\phi_{k+1,m}\rangle \|$. Now by the triangle inequality,

$$\begin{aligned}
\| |\phi_x\rangle - |\phi_y\rangle \| &= \| |\phi_{m,m}\rangle - |\phi_{0,m}\rangle \| \\
&\leqslant \sum_k \| |\phi_{k+1,m}\rangle - |\phi_{k,m}\rangle \| \\
&\leqslant \sum_k \sqrt{\sum_{j\in\varDelta(x,y)} q_j(|\phi_k\rangle)} \\
&\leqslant \sqrt{m}\sqrt{\sum_k \sum_{j\in\varDelta(x,y)} q_j(|\phi_k\rangle)} \\
&= \sqrt{m}\sqrt{\sum_{j\in\varDelta(x,y)} \sum_k = q_j(|\phi_k\rangle)} \\
&= \sqrt{m}\sqrt{\sum_{j\in\varDelta(x,y)} q_j(x)}.
\end{aligned}$$

■

Corollary 3.2. *If* $\sum_{j\in\varDelta(x,y)} q_j(x) \leqslant \epsilon^2/m$ *then* $\| |\phi_x\rangle - |\phi_y\rangle \| \leqslant \epsilon$.

Theorem 3.3. *Any quantum algorithm with error probability bounded by* $\frac{1}{3}$, *must make at least* $\frac{1}{12}\sqrt{n}$ *queries to decide whether or not its input is* 0^n.

Proof. Suppose that algorithm $\mathcal{A}$ solves this problem in m queries. Consider a run of $\mathcal{A}$ on input $x = 0^n$. Then the query magnitude of bit position j is $q_j(x)$. Notice that if j is chosen uniformly at random, then the expected value of this query magnitude, $E[q_j(x)] = m/n$. Let y be the n-bit string with a single 1 in the jth bit position (i.e. $y = 0^{j-1}10^{n-j}$. Therefore by the swapping lemma, the norm of the difference between the final state vectors on input x and y can be bounded by $\| |\phi_x\rangle - |\phi_y\rangle \| \leqslant \sqrt{m}\sqrt{m/n} = m/\sqrt{n}$. Now if $m < \frac{1}{12}\sqrt{n}$ then by lemma 2.1, $|\mathcal{D}(\phi_x) - \mathcal{D}(\phi_y)|_1 < \frac{1}{3}$. Since this contradicts the bound on the error probability of $\mathcal{A}$, it follows that $m \geqslant \frac{1}{12}\sqrt{n}$. ■

Corollary 3.4. *There is an oracle A such that* $\mathrm{NP}^A \not\subseteq \mathrm{BQTIME}(o(2^{n/2}))^A$

Note. The proof of corollary 3.4 follows easily from theorem 3.3, and standard diagonalization arguments from complexity theory. Therefore we omit its proof.

4. Block sensitivity and the black-box model

In this section we continue to work in the black-box model. Recently, Beals *et al.* (1998) introduced a new framework, involving polynomials, for establishing lower bounds in the black-box model; they proved a general result showing that $D(f) = O(Q(f)^6)$. We sketched a proof of this result. First let us define $D(f)$ and $Q(f)$.

For a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, denote by $Q(f)$ the quantum complexity of computing f with error probability at most $\frac{1}{3}$ in the black-box model, i.e. $Q(f)$ is the minimum number of queries to the input, x, that a quantum algorithm must make to compute $f(x)$ with error probability at most $\frac{1}{3}$.

Denote by $D(f)$ the deterministic complexity of f in the black-box model, i.e. the minimum number of bits of the input that a deterministic algorithm must query to compute f.

Denote by $C(f)$ the certificate complexity or the non-deterministic complexity of f in the black-box model, i.e. the minimum number of bits of the input that must be revealed (by someone who knows all the input bits) to convince a deterministic algorithm about the value of $f(x)$.

A key result, that was first discovered by Blum & Impagliazzo (1987), shows that in the black-box model the deterministic and non-deterministic (certificate) complexity of a function are polynomially related. Recall that in the black-box model we only count the number of queries made by the algorithm, not the number of steps of computation performed by the algorithm between queries. In fact, the $C(f)^2$ upper bound on the deterministic complexity is established by giving an algorithm that requires $2^{C(f)}$ steps of computation, but only $C(f)^2$ queries.

Lemma 4.1 (Blum & Impagliazzo 1987). $C(f) \leqslant D(f) \leqslant C(f)^2$.

Nisan (1989) established another fundamental result, that shows that the black-box complexity of a Boolean function f is closely related to a structural property of f called its block sensitivity. To define this notion, we need some notation.

Notation. For a string $x \in \{0,1\}^n$, and a set $S =\subseteq \{1,2,\dots,n\}$, define $x^{(S)}$ to be the boolean string y that differs from $x =$ on exactly the bit positions in the set S.

Definition. For a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ the block sensitivity of f, $bs(f)$ is defined to be the maximum number t such that there exists an input $x \in \{0,1\}^n$ and t disjoint subsets $S_1, \dots S_t \subseteq \{1,2,\dots,n\}$ such that for all $1 \leqslant i \leqslant t$, $f(x) \neq f(x^{(S_i)})$.

Lemma 4.2 (Nisan 1989). $\sqrt{C(f)} \leqslant bs(f) \leqslant C(f)$.

Corollary 4.3. $bs(f) \leqslant C(f) \leqslant D(f) \leqslant C(f)^2 \leqslant bs(f)^4$.

Beals *et al.* (1998) improve this bound by showing:

Lemma 4.4. $D(f) \leqslant C(f)bs(f) \leqslant bs(f)^3$.

We are now ready to prove the main result of this section.

Theorem 4.5. $Q(f) \geqslant \frac{1}{12}\sqrt{bs(f)}$.

Proof. Let $x \in \{0,1\}^n$ be the string such that there exist $t = bs(f)$ disjoint subsets $S_1, \dots S_t \subseteq \{1,2,\dots,n\}$ such that for all $1 \leqslant i \leqslant t$, $f(x) \neq f(x^{(S_i)})$. Given a quantum algorithm $\mathcal{A}$ let us define the query magnitude of set $S \subseteq \{1,2,\dots,n\}$ on input x to be $q_S(x) = \sum_{j \in S} q_j(x)$. Now if $\mathcal{A}$ runs for m steps, then since the sets S_i are disjoint, the expected query magnitude for a random set S_i is $q_{S_i}(x) \leqslant m/\,bs(f)$. Let $y = x^{(S_i)}$.

Therefore by the swapping lemma, the norm of the difference between the final state vectors on input x and y can be bounded by $\||\phi_x\rangle - |\phi_y\rangle\| \leqslant \sqrt{m}\sqrt{m/t} = m/\sqrt{t}$. Now if $m < \frac{1}{12}\sqrt{n}$ then by lemma 2.1, $|\mathcal{D}|\phi_x\rangle - \mathcal{D}|\phi_y\rangle|_1 < \frac{1}{3}$. Since this contradicts the bound on the error probability of $\mathcal{A}$, it follows that $m \geqslant \frac{1}{12}\sqrt{n}$. ■

Corollary 4.6. *$Q(f) \geqslant \frac{1}{12}D(f)^{1/6}$. Therefore $D(f) = O(Q(f)^6)$.*

I thank Leonard Schulman for his insightful comments on a draft of the paper. The author was supported by a JSEP grant.

References

Aharanov, D. & Ben-Or, M. 1996 Fault tolerant quantum computation with constant error. quant-ph/9611025.

Arora, S., Lund, C., Motwani, R., Sudan, M. & Szegedy, M. 1992 Proof verification and intractability of approximation problems. *Proc. 33rd A. Symp. on Foundations of Computer Science*, pp. 14–23.

Arora, S., Impagliazzo, R. & Vazirani, U. 1994 On the role of the Cook–Levin theorem in complexity theory. (Escript available at `http//www.cs.berkeley.edu/~vazirani`.)

Beals, R., Buhrman, H., Cleve, R., Mosca, M. & de Wolf, R. 1998 Quantum lower bounds by polynomials. quant-ph/9802049.

Bennett, C., Bernstein, E., Brassard, G. & Vazirani, U. 1997 Strengths and weaknesses of quantum computation. *SIAM Jl Comp.* (Special Issue on Quantum Computing, October.)

Bernstein, E. & Vazirani, U. 1993 Quantum complexity theory. *Proc. of the 25th A. ACM Symp. on Theory of Computing*, pp. 11–20.

Bernstein, E. & Vazirani, U. 1997 *SIAM J. Comp.* (Special Issue on Quantum Computing, October.)

Blum, M. & Impagliazzo, R. 1987 Generic oracles and oracle classes. In *28th A. Symp. on the Foundations of Computer Science*.

Boyer, M., Brassard, G., Høyer, P. & Tapp, A. 1996 Tight bounds on quantum searching. *Proc. 4th Workshop on Physics and Computation, Boston, 11/1996*, pp. 36–43. New England Complex Systems Institute. (Available at *InterJournal* at `http://interjournal.org`.)

Cook, S. A. 1971 The complexity of theorem-proving procedures. In *Proc. 3rd ACM Symp. on Theory of Computing*, pp. 151–158.

Deutsch, D. 1989 Quantum computational networks. *Proc. R. Soc. Lond.* A **425**, 73–90.

Deutsch, D. & Jozsa, R. 1992 Rapid solution of problems by quantum computation. *Proc. R. Soc. Lond.* A **439**, 553–558.

Gottesman, D. 1997 A theory of fault-tolerant quantum computation. quant-ph/9702029.

Grover, L. 1996 A fast quantum mechanical algorithm for database search. *Proc. 28th A. ACM Symp. on Theory of Computing*, pp. 212–219.

Levin, L. 1973 Universal'nyĭe perebornyĭe zadachi. *Problemy Peredachi Informatsii* **9**, 265–266 (In Russian).

Nisan, N. 1989 CREW PRAM's and decision trees. In *Proc. 21st ACM Symp. on Theory of Computing*, pp. 327–335.

Shamir, A. 1979 Factoring numbers in $O(\log n)$ arithmetic steps. *Informat. Processing Lett.* **8**(10), 28–31.

Shamir, A. 1990 IP = PSPACE. In *Proc. 22nd ACM Symp. on Theory of Computing*, pp. 11–15.

Shor, P. W. 1994 Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. 35th A. IEEE Symp. on Foundations of Computer Science*, pp. 124–134.

Shor, P. W. 1997 *SIAM Jl Comp.* (Special Issue on Quantum Computing, October.)

Simon, D. 1994 On the power of quantum computation. *Proc. 35th A. IEEE Symp. on Foundations of Computer Science*, pp. 116–123.

Simon, D. 1997 *SIAM Jl Comp.* (Special Issue on Quantum Computing, October.)

Vergis, A., Steiglitz, K. & Dickinson, B. 1986 The complexity of analog computation. *Math. Computers Simul.* **28**, 91–113.

Discussion

P. Marcer (*BCS Cybernetic Machine Group, Keynsham, UK*). If quantum theory is regarded as a Lie transformational system (as it is in relation to quantum holography), then the natural diffeomorphism, an exponential mapping, always has an analytic inverse. Quantum complexity theory is therefore in principle and via pattern matching or image processing able to deal with exponential towers of complexity in polynomial or even real time.

U. Vazirani. To properly evaluate the costs of quantum computation, we must consider a discrete model of computation. In such a model, quantum computers have at most an exponential factor complexity advantage over classical computers.