On the Power of Quantum Computation

Umesh Vazirani *
Computer Science Division,
University of California,
Berkeley, CA 94720.
Email: vazirani@cs.berkeley.edu

April 16, 1998

1 Abstract

This paper surveys the use of the 'hybrid argument' to prove to that quantum computations 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 quantum 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.

2 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

*Supported by a JSEP grant.

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 and Vazirani [BV], building upon earlier work of Deutsch and Jozsa [DJ]. This was followed by an important group theoretic quantum algorithm by Simon [Si], and then by the remarkable result of Shor [Sh] 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 $NP \cap co - NP$) in polynomial time. Bennett, et. al. [BBBV] 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, $NP \not\subseteq BQTIME(o(2^{n/2}))$. This result is best possible, up to constant factors, since there is a matching upper bound that follows from Grover's [Gr] quantum search algorithm. Boyer, et. al. [BBHT] obtain the exact constants in the upper and lower bounds, thus exhibiting a tight bound on quantum search. [BBBV] also showed that relative to a random permutation oracle $NP \cap co - NP \not\subseteq BQTIME(o(2^{n/3}))$.

As in all oracle lower bounds, the results of [BBBV] 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 $f : \{0,1\} \to \{0,1\}$, and let $Q(f)$ denote the number of queries made by a quantum algorithm that computes f with error probability less by $1/3$. Recently, Beals, et. al. [BBCMW] 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 [Sha1] 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 and Vazirani [BV] showed that in any quantum circuit with m gates, if each rotation gate is accurate to with in $\delta\theta \leq \epsilon/4m$ 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-tolerant quantum circuits can be created that are resilient to constant error in the rotation gates, independent of the size of the circuit [AB, Go, KLZ].

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 was 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 [Sha1, VSD]. 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 1 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 with in $1/k$ 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.

3 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 $|\theta' - \theta| \leq \delta\theta$. The corresponding unitary transformation U' satisfies $||U - U'|| \leq \delta\theta$. [1] How severely is the output of the circuit changed if each U_k is replaced by a perturbation V_k such that $||\, V_k - U_k \,|| \leq \delta$?

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 k^{th} one is perturbed. In this case, the state of the n qubits after the perturbed k^{th} 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 k^{th} 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

[1] The norm of a linear operator A is

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

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

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

$$
\begin{aligned}
|| \, |\psi'\rangle - |\psi\rangle \, || &= || \, U_m \cdots U_{k+1} U_k U_{k-1} \cdots U_1 \, |\phi_0\rangle - U_m \cdots U_{k+1} V_k U_{k-1} \cdots U_1 \, |\phi_0\rangle \, || \\
&= || \, U_m \cdots U_{k+1} (U_k - V_k) U_{k-1} \cdots U_1 \, |\phi_0\rangle \, || \\
&= || \, (U_k - V_k) \, |\phi_{k-1}\rangle \, || \\
&\leq || \, V_k - U_k \, ||
\end{aligned}
$$

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 [BV] 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 1 *If* $|| \, |\psi'\rangle - |\psi\rangle \, || \leq \epsilon$, *then* $|| \, \mathcal{D}(\psi') - \mathcal{D}(\psi) \, ||_1 \leq 4\epsilon$.

Now consider the general case where each gate U_k is replaced by a perturbation V_k such that $||U_k - V_k|| \leq \delta$. 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 j^{th} run, the last j gates are accurate, but the first $m - j$ gates are replaced by their perturbations. Denote by $|\psi_j\rangle$ the final state vector on the j^{th} 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 $|| \, |\psi_0\rangle - |\psi_m\rangle \, || \leq m\delta$. Finally, appealing to lemma 1, if $\delta \leq \frac{\epsilon}{4m}$, then the output distribution of the ideal and the perturbed circuit differ by at most ϵ.

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

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	$\cdots$	V_2	U_2	U_2
3	V_3	V_3	V_3	V_3		U_3	U_3	U_3
$\vdots$		$\vdots$			$\cdots$		$\vdots$	
$m-2$	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}	$\cdots$	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	$\lvert\psi_0\rangle$	$\lvert\psi_1\rangle$	$\lvert\psi_2\rangle$	$\lvert\psi_3\rangle$	$\cdots$	$\lvert\psi_{m-2}\rangle$	$\lvert\psi_{m-1}\rangle$	$\lvert\psi_m\rangle$

Figure 1: The Hybrid Argument.

4 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 look at. 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, 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 $NP^A \not\subseteq BQTIME(o(2^{n/2}))^A$. The actual results of [BBBV] on which this section is based are somewhat stronger. They show that that relative to a random oracle, $NP \not\subseteq BQTIME(o(2^{n/2}))$. They also show that relative to a random permutation oracle $NP \cap co-NP \not\subseteq BQTIME(o(2^{n/3}))$.

How significant are these oracle results in view of the recent non-relativizing results in complexity theory, such as $IP = PSPACE$ [Sha2], and $NP = PCP(\log n, 1)$ [ALMSS]? Arora, et. al. [AIV] 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 [Co, Le]. The non-relativizing form of the Cook-Levin theorem says that any language with a polynomial time checkable proof of membership also has a checkable proof of membership. [AIV] 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 [BBBV] indicate that there are two ways to design a polynomial time quantum algorithm for NP (or $NP \cap co-NP$): 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 analog 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. [BBHT].

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 the output of $\mathcal{A}$ is 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 j^{th} 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 2 (Swapping Lemma) *Let $|\phi_x\rangle$ and $|\phi_y\rangle$ denote the final states of $\mathcal{A}$ on inputs x and y respectively. Then*

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

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 k^{th} 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 k^{th} 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 k^{th} 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 k^{th} 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 $k+1^{st}$ query on the k^{th} 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 $j \in \Delta(x,y)$, it follows that $|| \, |\phi_{k,k+1}\rangle - |\phi_{k+1,k+1}\rangle \, ||^2 \leq \sum_{j \in \Delta(x,y)} q_j(|\phi_k\rangle)$. 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:

$$\begin{aligned} || \, |\phi_x\rangle - |\phi_y\rangle \, || &= || \, |\phi_{m,m}\rangle - |\phi_{0,m}\rangle \, || \\ &\leq \sum_k || \, |\phi_{k+1,m}\rangle - |\phi_{k,m}\rangle \, || \end{aligned}$$

$$
\begin{aligned}
&\le \sum_k \sqrt{\sum_{j \in \Delta(x,y)} q_j(|\phi_k\rangle)} \\
&\le \sqrt{m} \sqrt{\sum_k \sum_{j \in \Delta(x,y)} q_j(|\phi_k\rangle)} \\
&= \sqrt{m} \sqrt{\sum_{j \in \Delta(x,y)} \sum_k q_j(|\phi_k\rangle)} \\
&= \sqrt{m} \sqrt{\sum_{j \in \Delta(x,y)} q_j(x)}.
\end{aligned}
$$

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

Theorem 3 *Any quantum algorithm with error probability bounded by* $1/3$, *must make at least* $\sqrt{n}/12$ *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 j^{th} 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: $|| \, |\phi_x\rangle - |\phi_y\rangle \, || \le \sqrt{m}\sqrt{m/n} = m/\sqrt{n}$. Now if $m < \sqrt{n}/12$ then by Lemma 1, $|\mathcal{D}(\phi_x) - \mathcal{D}(\phi_y)|_1 < 1/3$. Since this contradicts the bound on the error probability of $\mathcal{A}$, it follows that $m \ge \sqrt{n}/12$.

Corollary 4 *There is an oracle* A *such that* $NP^A \not\subseteq BQTIME(o(2^{n/2}))^A$

Note: The proof of Corollary 4 follows easily from Theorem 3, and standard diagonalization arguments from complexity theory. Therefore we omit its proof.

5 Block Sensitivity and the Black Box Model:

In this section we continue to work in the black box model. Recently, Beals, et. al. [BBCMW] 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 sketch a proof of this result. First let us define $D(f)$ and $Q(f)$:

For a boolean function $f : \{0,1\}^n \to \{0,1\}$, denote by $Q(f)$ the quantum complexity of computing f with error probability at most 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 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 nondeterministic 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 [BI], shows that in the black box model the deterministic and nondeterministic (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 3 (BI) $C(f) \le D(f) \le C(f)^2$.

Nisan [Ni] 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 $S \subseteq \{1, 2, \ldots, n\}$, 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 $f : \{0,1\}^n \to \{0,1\}$ 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 $S_1, \ldots S_t \subseteq \{1, 2, \ldots, n\}$ such that for all $1 \le i \le t$, $f(x) \neq f(x^{(S_i)})$.

Lemma 4 (Ni) $\sqrt{C(f)} \le bs(f) \le C(f)$.

Corollary 5 $bs(f) \le C(f) \le D(f) \le C(f)^2 \le bs(f)^4$.

[BBCMW] improve this bound by showing:

Lemma 5 $D(f) \le C(f)bs(f) \le bs(f)^3$.

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

Theorem 6 $Q(f) \ge \sqrt{bs(f)}/12$.

Proof: Let $x \in \{0,1\}^n$ be the string such that there exist $t = bs(f)$ disjoint subsets $S_1, \ldots S_t \subseteq \{1, 2, \ldots, n\}$ such that for all $1 \le i \le t$, $f(x) \neq f(x^{(S_i)})$. Given a quantum algorithm $\mathcal{A}$ let us define the query magnitude of set $S \subseteq \{1, 2, \ldots, n\}$ 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 $q_{S_i}(x) \le m/bs(f)$. 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: $|| \, |\phi_x\rangle - |\phi_y\rangle \, || \le \sqrt{m}\sqrt{m/k} = m/\sqrt{k}$. Now if $m < \sqrt{n}/12$ then by Lemma 1, $|\mathcal{D}|\phi_x\rangle - \mathcal{D}|\phi_y\rangle|_1 < 1/3$. Since this contradicts the bound on the error probability of $\mathcal{A}$, it follows that $m \ge \sqrt{k}/12$.

Corollary 7 $Q(f) \ge D(f)^{1/6}/12$. *Therefore* $D(f) = O(Q(f)^6)$.

Acknowledgements: I am grateful to Leonard Schulman for his insightful comments on a draft of the paper.

References

[AB] Aharanov, D., and Ben-Or, M., "Fault tolerant quantum computation with constant error", quant-ph/9611025.

[AIV] Arora, S., Impagliazzo, R. and Vazirani, U., "On the Role of the Cook-Levin Theorem in Complexity Theory", manuscript (1994).

[ALMSS] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. "Proof verification and intractability of approximation problems". *33rd Annual Symposium on Foundations of Computer Science*, IEEE computer society press, pages 14-23, 1992.

[BBBV] Bennett, C., Bernstein, E., Brassard, G. and Vazirani, U., "Strengths and Weaknesses of Quantum Computation", *Siam Journal of Computing*, **26**, October, 1997 (special issue on quantum computation).

[BBCMW] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, Ronald de Wolf "Quantum Lower Bounds by Polynomials", quant-ph/9802049.

[BI] Blum, M. and Impagliazzo, R., "Generic oracles and oracle classes", 28th Annual Symposium on Foundations of Computer Science, IEEE computer society press, 1987.

[BV] Bernstein, E. and Vazirani, U., "Quantum complexity theory", *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, 1993, pp. 11 – 20. *Siam Journal of Computing*, **26**, October, 1997 (special issue on quantum computation).

[BBHT] Boyer, M., Brassard, G., Høyer, P. and Tapp, A., "Tight bounds on quantum searching", *Proceedings of the Fourth Workshop on Physics and Computation*, Boston, November 1996, New England Complex Systems Institute, pp. 36 – 43. Available online in the *InterJournal* at `http://interjournal.org`.

[Co] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[De2] Deutsch, D., "Quantum computational networks", *Proceedings of the Royal Society, London*, vol. A425, 1989, pp. 73 − 90.

[DJ] Deutsch, D. and Jozsa, R., "Rapid solution of problems by quantum computation", *Proceedings of the Royal Society, London*, vol. A439, 1992, pp. 553 − 558.

[Go] Gottesman, D., "A theory of fault-tolerant quantum computation", quant-ph/9702029.

[Gr] Grover, L., "A fast quantum mechanical algorithm for database search", *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 1996, pp. 212 − 219.

[KLZ] Knill, E., Laflamme, R., Zurek, W. "Resilient quantum computation: error models and thresholds", quant-ph/9702058.

[Le] L. Levin. Universal'nyĭe perebornyĭe zadachi (universal search problems : in russian). *Problemy Peredachi Informatsii*, 9(3):265–266, 1973.

[Ni] Nisan, N., "CREW PRAM's and decision trees", STOC 1989, pages 327-335.

[Sha1] Shamir, A., "Factoring numbers in O(log n) arithmetic steps", Information Processing Letters, 8(1):28-31, January 1979.

[Sha2] A. Shamir. IP=PSPACE. In *Proc. 22nd Annual ACM Symposium on Theory of Computing*, pages 11–15, 1990.

[Sh] Shor, P.W., "Algorithms for quantum computation: Discrete logarithms and factoring", *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994, pp. 124 − 134. *Siam Journal of Computing*, **26**, October, 1997, pp. 1484-1509 (special issue on quantum computation).

[Sh2] Shor, P. W., "Fault Tolerant Quantum Computation", *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, 1996, pp. 56 – 65.

[Si] Simon, D., "On the power of quantum computation", *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994, pp. 116 – 123. Special issue of *Siam J. Comp.*, October, 1997.

[VSD] Vergis, A., Steiglitz, K., Dickinson, B., "The complexity of analog computation", Mathematics and Computers in Simulation 28 (1986) 91-113.