Average-Case Quantum Query Complexity

Andris Ambainis[1]* and Ronald de Wolf[2,3]

[1] Computer Science Department, University of California, Berkeley CA 94720,
ambainis@cs.berkeley.edu
[2] CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands,
rdewolf@cwi.nl
[3] ILLC, University of Amsterdam

Abstract. We compare classical and quantum query complexities of total Boolean functions. It is known that for *worst-case* complexity, the gap between quantum and classical can be at most polynomial [3]. We show that for *average-case* complexity under the uniform distribution, quantum algorithms can be exponentially faster than classical algorithms. Under non-uniform distributions the gap can even be super-exponential. We also prove some general bounds for average-case complexity and show that the average-case quantum complexity of MAJORITY under the uniform distribution is nearly quadratically better than the classical complexity.

1 Introduction

The field of quantum computation studies the power of computers based on quantum mechanical principles. So far, most quantum algorithms—and *all* physically implemented ones—have operated in the so-called *black-box* setting. Examples are [9,18,11,7,8]; even period-finding, which is the core of Shor's factoring algorithm [17], can be viewed as a black-box problem. Here the input of the function f that we want to compute can only be accessed by means of queries to a "black-box". This returns the ith bit of the input when queried on i. The complexity of computing f is measured by the required number of queries. In this setting we want quantum algorithm that use significantly fewer queries than the best classical algorithms.

We restrict attention to computing total Boolean functions f on N variables. The query complexity of f depends on the kind of errors one allows. For example, we can distinguish between exact computation, zero-error computation (a.k.a. Las Vegas), and bounded-error computation (Monte Carlo). In each of these models, *worst-case* complexity is usually considered: the complexity is the number of queries required for the "hardest" input. Let $D(f)$, $R(f)$ and $Q(f)$ denote the worst-case query complexity of computing f for classical deterministic algorithms, classical randomized bounded-error algorithms, and quantum bounded-error algorithms, respectively. Clearly $Q(f) \leq R(f) \leq D(f)$.

* Part of this work was done when visiting Microsoft Research.

H. Reichel and S. Tison (Eds.): STACS 2000, LNCS 1770, pp. 133–144, 2000.

The main quantum success here is Grover's algorithm [11]. It can compute the OR-function with bounded-error using $\Theta(\sqrt{N})$ queries (this is optimal [4,5,20]). Thus $Q(\mathrm{OR}) \in \Theta(\sqrt{N})$, whereas $D(\mathrm{OR}) = N$ and $R(\mathrm{OR}) \in \Theta(N)$. This is the biggest gap known between quantum and classical worst-case complexities for total functions. (In contrast, for *partial* Boolean functions the gap can be much bigger [9,18].) A recent result is that the gap between $D(f)$ and $Q(f)$ is at most polynomial for *every* total f: $D(f) \in O(Q(f)^6)$ [3]. This is similar to the best-known relation between classical deterministic and randomized algorithms: $D(f) \in O(R(f)^3)$ [16].

Given some probability distribution μ on the set of inputs $\{0,1\}^N$ one may also consider *average-case* complexity instead of worst-case complexity. Average-case complexity concerns the expected number of queries needed when the input is distributed according to μ. If the hard inputs receive little μ-probability, then average-case complexity can be significantly smaller than worst-case complexity. Let $D^\mu(f)$, $R^\mu(f)$, and $Q^\mu(f)$ denote the average-case analogues of $D(f)$, $R(f)$, and $Q(f)$, respectively. Again $Q^\mu(f) \leq R^\mu(f) \leq D^\mu(f)$. The objective of this paper is to compare these measures and to investigate the possible gaps between them. Our main results are:

- Under uniform μ, $Q^\mu(f)$ and $R^\mu(f)$ can be super-exponentially smaller than $D^\mu(f)$.
- Under uniform μ, $Q^\mu(f)$ can be exponentially smaller than $R^\mu(f)$. Thus the [3]-result for worst-case quantum complexity does not carry over to the average-case setting.
- Under non-uniform μ the gap can be even larger: we give distributions μ where $Q^\mu(\mathrm{OR})$ is constant, whereas $R^\mu(\mathrm{OR})$ is almost $\sqrt{N}$. (Both this gap and the previous one still remains if we require the quantum algorithm to work with zero-error instead of bounded-error.)
- For every f and μ, $R^\mu(f)$ is lower bounded by the expected *block sensitivity* $E_\mu[bs(f)]$ and $Q^\mu(f)$ is lower bounded by $E_\mu[\sqrt{bs(f)}]$.
- For the MAJORITY-function under uniform μ, we have $Q^\mu(f) \in O(N^{1/2+\varepsilon})$ for every $\varepsilon > 0$, and $Q^\mu(f) \in \Omega(N^{1/2})$. In contrast, $R^\mu(f) \in \Omega(N)$.
- For the PARITY-function, the gap between Q^μ and R^μ can be quadratic, but not more. Under uniform μ, PARITY has $Q^\mu(f) \in \Omega(N)$.

2 Definitions

Let $f : \{0,1\}^N \to \{0,1\}$ be a Boolean function. It is *symmetric* if $f(X)$ only depends on $|X|$, the Hamming weight (number of 1s) of X. $\overline{0}$ denotes the input with weight 0. We will in particular consider the following functions: $\mathrm{OR}(X) = 1$ iff $|X| \geq 1$; $\mathrm{MAJ}(X) = 1$ iff $|X| > N/2$; $\mathrm{PARITY}(X) = 1$ iff $|X|$ is odd. If $X \in \{0,1\}^N$ is an input and S a set of (indices of) variables, we use X^S to denote the input obtained by flipping the values of the S-variables in X. The *block sensitivity* $bs_X(f)$ of f on input X is the maximal number b for which there are b disjoint sets of variables $S_1, \ldots, S_b$ such that $f(X) \neq f(X^{S_i})$ for all $1 \leq i \leq b$. The block sensitivity $bs(f)$ of f is $\max_X bs_X(f)$.

We focus on three kinds of algorithms for computing f: classical *deterministic*, classical *randomized* bounded-error, and *quantum* bounded-error algorithms. If A is an algorithm (quantum or classical) and $b \in \{0,1\}$, we use $\Pr[A(X) = b]$ to denote the probability that A answers b on input X. We use $T_A(X)$ for the expected number of queries that A uses on input X.[1] Note that this only depends on A and X, not on the input distribution μ. For deterministic A, $\Pr[A(X) = b] \in \{0,1\}$ and the expected number of queries $T_A(X)$ is the same as the actual number of queries.

Let $\mathcal{D}(f)$ denote the set of classical *deterministic* algorithms that compute f. Let $\mathcal{R}(f) = \{\text{classical } A \mid \forall X \in \{0,1\}^N : \Pr[A(X) = f(X)] \geq 2/3\}$ be the set of classical *randomized* algorithms that compute f with bounded error probability. Similarly let $\mathcal{Q}(f)$ be the set of *quantum* algorithms that compute f with bounded-error. We define the following *worst-case complexities*:

$$D(f) = \min_{A \in \mathcal{D}(f)} \max_{X \in \{0,1\}^N} T_A(X)$$

$$R(f) = \min_{A \in \mathcal{R}(f)} \max_{X \in \{0,1\}^N} T_A(X)$$

$$Q(f) = \min_{A \in \mathcal{Q}(f)} \max_{X \in \{0,1\}^N} T_A(X)$$

$D(f)$ is also known as the *decision tree complexity* of f and $R(f)$ as the *bounded-error* decision tree complexity of f. Since quantum generalizes randomized and randomized generalizes deterministic computation, we have $Q(f) \leq R(f) \leq D(f)$ for all f. The three worst-case complexities are polynomially related: $D(f) \in O(R(f)^3)$ [16] and $D(f) \in O(Q(f)^6)$ [3] for all total f.

Let $\mu : \{0,1\}^N \to [0,1]$ be a probability distribution. We define the *average-case complexity* of an algorithm A with respect to a distribution μ as:

$$T_A^\mu = \sum_{X \in \{0,1\}^N} \mu(X) T_A(X).$$

The average-case deterministic, randomized, and quantum complexities of f with respect to μ are

$$D^\mu(f) = \min_{A \in \mathcal{D}(f)} T_A^\mu$$

$$R^\mu(f) = \min_{A \in \mathcal{R}(f)} T_A^\mu$$

$$Q^\mu(f) = \min_{A \in \mathcal{Q}(f)} T_A^\mu$$

Note that the algorithms still have to output the correct answer on *all* inputs, even on X that have $\mu(X) = 0$. Clearly $Q^\mu(f) \leq R^\mu(f) \leq D^\mu(f)$ for all μ and

[1] See [3] for definitions and references for the quantum circuit model. A satisfactory formal definition of *expected* number of queries $T_A(X)$ for a quantum algorithm A is a hairy issue, involving the notion of a stopping criterion. We will not give such a definition here, since in the bounded-error case, expected and worst-case number of queries can be made the same up to a small constant factor.

f. Our goal is to examine how large the gaps between these measures can be, in particular for the uniform distribution $unif(X) = 2^{-N}$.

The above treatment of average-case complexity is the standard one used in average-case analysis of algorithms [19]. One counter-intuitive consequence of these definitions, however, is that the average-case performance of polynomially related algorithms can be superpolynomially apart (we will see this happen in Section 5). This seemingly paradoxical effect makes these definitions unsuitable for dealing with polynomial-time reducibilities and average-case complexity classes, which is what led Levin to his alternative definition of "polynomial time on average" [13].[2] Nevertheless, we feel the above definitions are the appropriate ones for our query complexity setting: they just *are* the average number of queries that one needs when the input is drawn according to distribution μ.

3 Super-Exponential Gap between $D^{unif}(f)$ and $Q^{unif}(f)$

Here we show that $D^{unif}(f)$ can be much larger then $R^{unif}(f)$ and $Q^{unif}(f)$:

Theorem 1. *Define f on N variables such that $f(X) = 1$ iff $|X| \geq N/10$. Then $Q^{unif}(f)$ and $R^{unif}(f)$ are $O(1)$ and $D^{unif}(f) \in \Omega(N)$.*

Proof. Suppose we randomly sample k bits of the input. Let $a = |X|/N$ denote the fraction of 1s in the input and $\tilde{a}$ the fraction of 1s in the sample. Standard Chernoff bounds imply that there is a constant $c > 0$ such that

$$\Pr[\tilde{a} < 2/10 \mid a \geq 3/10] \leq 2^{-ck}.$$

Now consider the following randomized algorithm for f:

1. Let $i = 1$.
2. Sample $k_i = i/c$ bits. If the fraction $\tilde{a}_i$ of 1s is $\geq 2/10$, output 1 and stop.
3. If $i < \log N$, increase i by 1 and repeat step 2.
4. If $i \geq \log N$, count N exactly using N queries and output the correct answer.

It is easily seen that this is a bounded-error algorithm for f. Let us bound its average-case complexity under the uniform distribution.

If $a \geq 3/10$, the expected number of queries for step 2 is

$$\sum_{i=1}^{\log N} \Pr[\tilde{a}_1 \leq 2/10, \ldots, \tilde{a}_{i-1} \leq 2/10 \mid a > 3/10] \cdot \frac{i}{c} \leq$$

$$\sum_{i=1}^{\log N} \Pr[\tilde{a}_{i-1} \leq 2/10 \mid a > 3/10] \cdot \frac{i}{c} \leq \sum_{i=1}^{\log N} 2^{-(i-1)} \cdot \frac{i}{c} \in O(1).$$

The probability that step 4 is needed (given $a \geq 3/10$) is at most $2^{-c \log N/c} = 1/N$. This adds $\frac{1}{N} N = 1$ to the expected number of queries.

[2] We thank Umesh Vazirani for drawing our attention to this.

The probability of $a < 3/10$ is $2^{-c'N}$ for some constant c'. This case contributes at most $2^{-c'N}(N + (\log N)^2) \in o(1)$ to the expected number of queries. Thus in total the algorithm uses $O(1)$ queries on average, hence $R^{unif}(f) \in O(1)$.

It is easy to see that any deterministic classical algorithm for f must make at least $N/10$ queries on every input, hence $D^{unif}(f) \geq N/10$. □

Accordingly, we can have huge gaps between $D^{unif}(f)$ and $Q^{unif}(f)$. However, this example tells us nothing about the gaps between quantum and classical bounded-error algorithms. In the next section we exhibit an f where $Q^{unif}(f)$ is exponentially smaller than $R^{unif}(f)$.

4 Exponential Gap between $R^{unif}(f)$ and $Q^{unif}(f)$

4.1 The Function

We use the following modification of Simon's problem [18]:[3]

Input: $X = (x_1, \ldots, x_{2^n})$, where each $x_i \in \{0,1\}^n$.
Output: $f(X) = 1$ iff there is a non-zero $k \in \{0,1\}^n$ such that $x_{i \oplus k} = x_i \ \forall i$.

Here we treat $i \in \{0,1\}^n$ both as an n-bit string and as a number, and $\oplus$ denotes bitwise XOR. Note that this function is total (unlike Simon's). Formally, f is not a Boolean function because the variables are $\{0,1\}^n$-valued. However, we can replace every variable x_i by n Boolean variables and then f becomes a Boolean function of $N = n2^n$ variables. The number of queries needed to compute the Boolean function is at least the number of queries needed to compute the function with $\{0,1\}^n$-valued variables (because we can simulate a query to the Boolean oracle with a query to the $\{0,1\}^n$-valued oracle by just throwing away the rest of the information) and at most n times the number of queries to the $\{0,1\}^n$-valued oracle (because one $\{0,1\}^n$-valued query can be simulated using n Boolean queries). As the numbers of queries are so closely related, it does not make a big difference whether we use the $\{0,1\}^n$-valued oracle or the Boolean oracle. For simplicity we count queries to the $\{0,1\}^n$-valued oracle.

The main result is the following exponential gap:

Theorem 2. *For f as above, $Q^{unif}(f) \leq 22n + 1$ and $R^{unif}(f) \in \Omega(2^{n/2})$.*

4.2 Quantum Upper Bound

The quantum algorithm is similar to Simon's. Start with the 2-register superposition $\sum_{i \in \{0,1\}^n} |i\rangle|0\rangle$ (for convenience we ignore normalizing factors). Apply the oracle once to obtain

$$\sum_{i \in \{0,1\}^n} |i\rangle|x_i\rangle.$$

[3] The recent preprint [12] proves a related but incomparable result about another modification of Simon's problem.

Measuring the second register gives some j and collapses the first register to

$$\sum_{i:x_i=j} |i\rangle.$$

Applying a Hadamard transform H to each qubit of the first register gives

$$\sum_{i:x_i=j} \sum_{i' \in \{0,1\}^n} (-1)^{(i,i')} |i'\rangle. \tag{1}$$

(a,b) denotes inner product mod 2; if $(a,b)=0$ we say a and b are orthogonal.

If $f(X)=1$, then there is a non-zero k such that $x_i = x_{i\oplus k}$ for all i. In particular, $x_i = j$ iff $x_{i\oplus k} = j$. Then the final state (1) can be rewritten as

$$\sum_{i' \in \{0,1\}^n} \sum_{i:x_i=j} (-1)^{(i,i')} |i'\rangle = \sum_{i' \in \{0,1\}^n} \left(\sum_{i:x_i=j} \frac{1}{2}((-1)^{(i,i')} + (-1)^{(i\oplus k,i')}) \right) |i'\rangle$$

$$= \sum_{i' \in \{0,1\}^n} \left(\sum_{i:x_i=j} \frac{(-1)^{(i,i')}}{2}(1 + (-1)^{(k,i')}) \right) |i'\rangle.$$

Notice that $|i'\rangle$ has non-zero amplitude only if $(k,i')=0$. Hence if $f(X)=1$, then measuring the final state gives some i' orthogonal to the unknown k.

To decide if $f(X)=1$, we repeat the above process $m = 22n$ times. Let $i_1, \ldots, i_m \in \{0,1\}^n$ be the results of the m measurements. If $f(X)=1$, there must be a non-zero k that is orthogonal to all i_r. Compute the subspace $S \subseteq \{0,1\}^n$ that is generated by $i_1, \ldots, i_m$ (i.e. S is the set of binary vectors obtained by taking linear combinations of $i_1, \ldots, i_m$ over $GF(2)$). If $S = \{0,1\}^n$, then the only k that is orthogonal to all i_r is $k = 0^n$, so then we know that $f(X)=0$. If $S \neq \{0,1\}^n$, we just query all 2^n values $x_{0\ldots0}, \ldots, x_{1\ldots1}$ and then compute $f(X)$. This latter step is of course very expensive, but it is needed only rarely:

Lemma 1. *Assume that $X = (x_{0\ldots0}, \ldots, x_{1\ldots1})$ is chosen uniformly at random from $\{0,1\}^N$. Then, with probability at least $1-2^{-n}$, $f(X)=0$ and the measured $i_1, \ldots, i_m$ generate $\{0,1\}^n$.*

Proof. It can be shown by a small modification of [1, Theorem 5.1, p.91] that with probability at least $1-2^{-c2^n}$ ($c>0$), there are at least $2^n/8$ values j such that $x_i = j$ for exactly one $i \in \{0,1\}^n$. We assume that this is the case.

If $i_1, \ldots, i_m$ generate a proper subspace of $\{0,1\}^n$, then there is a non-zero $k \in \{0,1\}^n$ that is orthogonal to this subspace. We estimate the probability that this happens. Consider some fixed non-zero vector $k \in \{0,1\}^n$. The probability that i_1 and k are orthogonal is at most $\frac{15}{16}$, as follows. With probability at least $1/8$, the measurement of the second register gives j such that $f(i) = j$ for a unique i. In this case, the measurement of the final superposition (1) gives a uniformly random i'. The probability that a uniformly random i' has $(k,i') \neq 0$ is $1/2$. Therefore, the probability that $(k,i_1)=0$ is at most $1 - \frac{1}{8} \cdot \frac{1}{2} = \frac{15}{16}$.

The vectors $i_1, \ldots, i_m$ are chosen independently. Therefore, the probability that k is orthogonal to each of them is at most $(\frac{15}{16})^{22n} < 2^{-2n}$. There are $2^n - 1$ possible non-zero k, so the probability that there is a k which is orthogonal to each of $i_1, \ldots, i_m$, is at most $(2^n - 1)2^{-2n} < 2^{-n}$. □

Note that this algorithm is actually a *zero-error* algorithm: it always outputs the correct answer. Its expected number of queries on a uniformly random input is at most $m = 22n$ for generating $i_1, \ldots, i_m$ and at most $\frac{1}{2^n}2^n = 1$ for querying all the x_i if the first step does not give $i_1, \ldots, i_m$ that generate $\{0,1\}^n$. This completes the proof of the first part of Theorem 2.

4.3 Classical Lower Bound

Let D_1 be the uniform distribution over all inputs $X \in \{0,1\}^N$ and D_2 be the uniform distribution over all X for which there is a unique $k \neq 0$ such that $x_i = x_{i \oplus k}$ (and hence $f(X) = 1$). We say an algorithm A *distinguishes* between D_1 and D_2 if the average probability that A outputs 0 is $\geq 3/4$ under D_1 and the average probability that A outputs 1 is $\geq 3/4$ under D_2.

Lemma 2. *If there is a bounded-error algorithm A that computes f with $m = T_A^{unif}$ queries on average, then there is an algorithm that distinguishes between D_1 and D_2 and uses $O(m)$ queries on all inputs.*

Proof. We run A until it stops or makes $4m$ queries. The average probability (under D_1) that it stops is at least $3/4$, for otherwise the average number of queries would be more than $\frac{1}{4}(4m) = m$. Under D_1, the probability that A outputs $f(X) = 1$ is at most $1/4 + o(1)$ ($1/4$ is the maximum probability of error on an input with $f(X) = 0$ and $o(1)$ is the probability of getting an input with $f(X) = 1$). Therefore, the probability under D_1 that A outputs 0 after at most $4m$ queries, is at least $3/4 - (1/4 + o(1)) = 1/2 - o(1)$.

In contrast, the D_2-probability that A outputs 0 is $\leq 1/4$ because $f(X) = 1$ for any input X from D_2. We can use this to distinguish D_1 from D_2. □

Lemma 3. *No classical randomized algorithm A that makes $m \in o(2^{n/2})$ queries can distinguish between D_1 and D_2.*

Proof. For a random input from D_1, the probability that all answers to m queries are different is

$$1 \cdot (1 - 1/2^n) \cdots (1 - (m-1)/2^n) \geq (1 - m/2^n)^m \to e^{-m^2/2^n} = 1 - o(1).$$

For a random input from D_2, the probability that there is an i s.t. A queries both x_i and $x_{i \oplus k}$ (k is the hidden vector) is $\leq \binom{m}{2}/(2^n - 1) \in o(1)$, since:

1. for every pair of distinct i, j, the probability that $i = j \oplus k$ is $1/(2^n - 1)$
2. since A queries only m of the x_i, it queries only $\binom{m}{2}$ distinct pairs i, j

If no pair x_i, $x_{i\oplus k}$ is queried, the probability that all answers are different is

$$1 \cdot (1 - 1/2^{n-1}) \cdots (1 - (m-1)/2^{n-1}) = 1 - o(1).$$

It is easy to see that all sequences of m different answers are equally likely. Therefore, for both distributions D_1 and D_2, we get a uniformly random sequence of m different values with probability $1-o(1)$ and something else with probability $o(1)$. Thus A cannot "see" the difference between D_1 and D_2 with sufficient probability to distinguish between them. □

The second part of Theorem 2 now follows: a classical algorithm that computes f with an average number of m queries can be used to distinguish between D_1 and D_2 with $O(m)$ queries (Lemma 2), but then $O(m) \in \Omega(2^{n/2})$ (Lemma 3).

5 Super-Exponential Gap for Non-uniform μ

The last section gave an exponential gap between Q^μ and R^μ under uniform μ. Here we show that the gap can be even larger for non-uniform μ. Consider the average-case complexity of the OR-function. It is easy to see that $D^{unif}(\mathrm{OR})$, $R^{unif}(\mathrm{OR})$, and $Q^{unif}(\mathrm{OR})$ are all $O(1)$, since the average input will have many 1s under the uniform distribution. Now we give some examples of non-uniform distributions μ where $Q^\mu(\mathrm{OR})$ is super-exponentially smaller than $R^\mu(\mathrm{OR})$:

Theorem 3. *If $\alpha \in (0,1/2)$ and $\mu(X) = c/\binom{N}{|X|}(|X|+1)^\alpha (N+1)^{1-\alpha}$ ($c \approx 1-\alpha$ is a normalizing constant), then $R^\mu(\mathrm{OR}) \in \Theta(N^\alpha)$ and $Q^\mu(\mathrm{OR}) \in \Theta(1)$.*

Proof. Any classical algorithm for OR requires $\Theta(N/(|X|+1))$ queries on input X. The upper bound follows from random sampling, the lower bound from a block-sensitivity argument [16]. Hence (omitting the intermediate Θs):

$$R^\mu(\mathrm{OR}) = \sum_X \mu(X) \frac{N}{|X|+1} = \sum_{t=0}^{N} \frac{cN^\alpha}{(t+1)^{\alpha+1}} \in \Theta(N^\alpha).$$

Similarly, for a quantum algorithm $\Theta(\sqrt{N/(|X|+1)})$ queries are necessary and sufficient on input X [11,5], so

$$Q^\mu(\mathrm{OR}) = \sum_X \mu(X) \sqrt{\frac{N}{|X|+1}} = \sum_{t=0}^{N} \frac{cN^{\alpha-1/2}}{(t+1)^{\alpha+1/2}} \in \Theta(1). \qquad \square$$

In particular, for $\alpha = 1/2 - \varepsilon$ we have the huge gap $O(1)$ quantum versus $\Omega(N^{1/2-\varepsilon})$ classical. Note that we obtain this super-exponential gap by weighing the complexity of two algorithms (classical and quantum OR-algorithms) which are only quadratically apart on each input X.

In fact, a small modification of μ gives the same big gap even if the quantum algorithm is forced to output the correct answer always. We omit the details.

6 General Bounds for Average-Case Complexity

In this section we prove some general bounds. First we make precise the intuitively obvious fact that if an algorithm A is faster on every input than another algorithm B, then it is also much faster on average under any distribution:

Theorem 4. *If $\phi : \mathbf{R} \to \mathbf{R}$ is a concave function and $T_A(X) \leq \phi(T_B(X))$ for all X, then $T_A^\mu \leq \phi(T_B^\mu)$ for every μ.*

Proof. By Jensen's inequality, if ϕ is concave then $E_\mu[\phi(T)] \leq \phi(E_\mu[T])$, hence

$$T_A^\mu \leq \sum_{X \in \{0,1\}^N} \mu(X)\phi(T_B(X)) \leq \phi\left(\sum_{X \in \{0,1\}^N} \mu(X) T_B(X)\right) = \phi\left(T_B^\mu\right). \qquad \square$$

In words: taking the average cannot make the complexity-gap between two algorithms smaller. For instance, if $T_A(X) \leq \sqrt{T_B(X)}$ (say, A is Grover's algorithm and B is a classical algorithm for OR), then $T_A^\mu \leq \sqrt{T_B^\mu}$. On the other hand, taking the average *can* make the gap much larger, as we saw in Theorem 3: the quantum algorithm for OR runs only quadratically faster than any classical algorithm on each input, but the *average-case* gap between quantum and classical can be much bigger than quadratic.

We now prove a general lower bound on R^μ and Q^μ. Using an argument from [16] for the classical case and an argument from [3] for the quantum case, we can show:

Lemma 4. *Let A be a bounded-error algorithm for some function f. If A is classical then $T_A(X) \in \Omega(bs_X(f))$, and if A is quantum then $T_A(X) \in \Omega(\sqrt{bs_X(f)})$.*

A lower bound in terms of the μ-expected block sensitivity follows:

Theorem 5. *For all f, μ: $R^\mu(f) \in \Omega(E_\mu[bs_X(f)])$ and $Q^\mu(f) \in \Omega(E_\mu[\sqrt{bs_X(f)}])$.*

7 Average-Case Complexity of MAJORITY

Here we examine the average-case complexity of the MAJORITY-function. The hard inputs for majority occur when $t = |X| \approx N/2$. Any quantum algorithm needs $\Omega(N)$ queries for such inputs [3]. Since the uniform distribution puts most probability on the set of X with $|X|$ close to $N/2$, we might expect an $\Omega(N)$ average-case complexity. However we will prove that the complexity is nearly $\sqrt{N}$. For this we need the following result about approximate quantum counting, which follows from [8, Theorem 5] (see also [14] or [15, Theorem 1.10]):

Theorem 6 (Brassard, Høyer, Tapp; Mosca). *Let $\alpha \in [0,1]$. There is a quantum algorithm with worst-case $O(N^\alpha)$ queries that outputs an estimate $\tilde{t}$ of the weight $t = |X|$ of its input, such that $|\tilde{t} - t| \leq N^{1-\alpha}$ with probability $\geq 2/3$.*

Theorem 7. *For every $\varepsilon > 0$, $Q^{unif}(\mathrm{MAJ}) \in O(N^{1/2+\varepsilon})$.*

Proof. Consider the following algorithm, with input X, and $\alpha \in [0,1]$ to be determined later.

1. Estimate $t = |X|$ by $\tilde{t}$ using $O(N^\alpha)$ queries.
2. If $\tilde{t} < N/2 - N^{1-\alpha}$ then output 0; if $\tilde{t} > N/2 + N^{1-\alpha}$ then output 1.
3. Otherwise use N queries to classically count t and output its majority.

It is easy to see that this is a bounded-error algorithm for MAJ. We determine its average complexity. The third step of the algorithm will be invoked iff $|\tilde{t} - N/2| \leq N^{1-\alpha}$. Denote this event by "$\tilde{t} \approx N/2$". For $0 \leq k \leq N^\alpha/2$, let D_k denote the event that $kN^{1-\alpha} \leq |t - N/2| < (k+1)N^{1-\alpha}$. Under the uniform distribution the probability that $|X| = t$ is $\binom{N}{t}2^{-N}$. By Stirling's formula this is $O(1/\sqrt{N})$, so the probability of the event D_k is $O(N^{1/2-\alpha})$. In the quantum counting algorithm, $\Pr[kN^{1-\alpha} \leq |\tilde{t} - t| < (k+1)N^{1-\alpha}] \in O(1/(k+1))$ (this follows from [6], the upcoming journal version of [8] and [14]). Hence also $\Pr[\tilde{t} \approx N/2 \mid D_k] \in O(1/(k+1))$. The probability that the second counting stage is needed is $\Pr[\tilde{t} \approx N/2]$, which we bound by

$$\sum_{k=0}^{N^\alpha/2} \Pr[\tilde{t} \approx N/2 \mid D_k] \cdot \Pr[D_k] = \sum_{k=0}^{N^\alpha/2} O(\frac{1}{k+1}) \cdot O(N^{1/2-\alpha}) = O(N^{1/2-\alpha} \log N).$$

Thus we can bound the average-case query complexity of our algorithm by

$$O(N^\alpha) + \Pr[\tilde{t} \approx N/2] \cdot N = O(N^\alpha) + O(N^{3/2-\alpha} \log N).$$

Choosing $\alpha = 3/4$, we obtain an $O(N^{3/4} \log N)$ algorithm.

However, we can reiterate this scheme: instead of using N queries in step 3 we could count using $O(N^{\alpha_2})$ instead of N queries, output an answer if there is a clear majority (i.e. $|\tilde{t} - N/2| > N^{1-\alpha_2}$), otherwise count again using $O(N^{\alpha_3})$ queries etc. If after k stages we still have no clear majority, we count using N queries. For any fixed k, we can make the error probability of each stage sufficiently small using only a constant number of repetitions. This gives a bounded-error algorithm for MAJORITY. (The above algorithm is the case $k = 1$.)

It remains to bound the complexity of the algorithm by choosing appropriate values for k and for the α_i (put $\alpha_1 = \alpha$). Let p_i denote the probability under *unif* that the ith counting-stage will be needed, i.e. that all previous counts gave results close to $N/2$. Then $p_{i+1} \in O(N^{1/2-\alpha_i} \log N)$ (as above). The average query complexity is now bounded by:

$$O(N^{\alpha_1}) + p_2 \cdot O(N^{\alpha_2}) + \cdots + p_k \cdot O(N^{\alpha_k}) + p_{k+1} \cdot N =$$

$$O(N^{\alpha_1}) + O(N^{1/2-\alpha_1+\alpha_2} \log N) + \cdots + O(N^{1/2-\alpha_{k-1}+\alpha_k} \log N) + O(N^{3/2-\alpha_k} \log N).$$

Clearly the asymptotically minimal complexity is achieved when all exponents in this expression are equal. This induces $k-1$ equations $\alpha_1 = 1/2 - \alpha_i + \alpha_{i+1}$, $1 \leq i < k$, and a kth equation $\alpha_1 = 3/2 - \alpha_k$. Adding up these k equations we obtain $k\alpha_1 = -\alpha_1 + (k-1)/2 + 3/2$, which implies $\alpha_1 = 1/2 + 1/(2k+2)$. Thus we have average query complexity $O(N^{1/2+1/(2k+2)} \log N)$. Choosing k sufficiently large, this becomes $O(N^{1/2+\varepsilon})$. □

The nearly matching lower bound is:

Theorem 8. $Q^{unif}(\text{MAJ}) \in \Omega(N^{1/2})$.

Proof. Let A be a bounded-error quantum algorithm for MAJORITY. It follows from the worst-case results of [3] that A uses $\Omega(N)$ queries on the hardest inputs, which are the X with $|X| = N/2 \pm 1$. Since the uniform distribution puts $\Omega(1/\sqrt{N})$ probability on the set of such X, the average-case complexity of A is at least $\Omega(1/\sqrt{N})\Omega(N) = \Omega(\sqrt{N})$. □

What about the *classical* average-case complexity? Alonso, Reingold, and Schott [2] prove that $D^{unif}(\text{MAJ}) = 2N/3 - \sqrt{8N/9\pi} + O(\log N)$. We can also prove that $R^{unif}(\text{MAJ}) \in \Omega(N)$ (for reasons of space we omit the details), so quantum is almost quadratically better than classical for this problem.

8 Average-Case Complexity of PARITY

Finally we prove some results for the average-case complexity of PARITY. This is in many ways the hardest Boolean function. Firstly, $bs_X(f) = N$ for all X, hence by Theorem 5:

Corollary 1. *For every* μ, $R^{\mu}(\text{PARITY}) \in \Omega(N)$ *and* $Q^{\mu}(\text{PARITY}) \in \Omega(\sqrt{N})$.

We can bounded-error quantum count $|X|$ exactly, using $O(\sqrt{(|X|+1)N})$ queries [8]. Combining this with a μ that puts $O(1/\sqrt{N})$ probability on the set of all X with $|X| > 1$, we obtain $Q^{\mu}(\text{PARITY}) \in O(\sqrt{N})$.

We can prove $Q^{\mu}(\text{PARITY}) \leq N/6$ for any μ by the following algorithm: with probability $1/3$ output 1, with probability $1/3$ output 0, and with probability $1/3$ run the exact quantum algorithm for PARITY, which has worst-case complexity $N/2$ [3,10]. This algorithm has success probability $2/3$ on every input and has expected number of queries equal to $N/6$.

More than a linear speed-up on average is not possible if μ is uniform:

Theorem 9. $Q^{unif}(\text{PARITY}) \in \Omega(N)$.

Proof. Let A be a bounded-error quantum algorithm for PARITY. Let B be an algorithm that flips each bit of its input X with probability $1/2$, records the number b of actual bitflips, runs A on the changed input Y, and outputs $A(Y) \oplus b$. It is easy to see that B is a bounded-error algorithm for PARITY and that it uses an expected number of T_A^{μ} queries on *every* input. Using standard techniques, we can turn this into an algorithm for PARITY with *worst-case* $O(T_A^{\mu})$ queries. Since the worst-case lower bound for PARITY is $N/2$ [3,10], the theorem follows. □

Acknowledgments

We thank Harry Buhrman for suggesting this topic, and him, Lance Fortnow, Lane Hemaspaandra, Hein Röhrig, Alain Tapp, and Umesh Vazirani for helpful discussions. Also thanks to Alain for sending a draft of [6].

References

1. N. Alon and J. H. Spencer. *The Probabilistic Method.* Wiley-Interscience, 1992.
2. L. Alonso, E. M. Reingold, and R. Schott. The average-case complexity of determining the majority. *SIAM Journal on Computing*, 26(1):1–14, 1997.
3. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. In *Proceedings of 39th FOCS*, pages 352–361, 1998. http://xxx.lanl.gov/abs/quant-ph/9802049.
4. C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. quant-ph/9701001.
5. M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–505, 1998. Earlier version in Physcomp'96. quant-ph/9605034.
6. G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. Forthcoming.
7. G. Brassard, P. Høyer, and A. Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News (Cryptology Column)*, 28:14–19, 1997. quant-ph/9705002.
8. G. Brassard, P. Høyer, and A. Tapp. Quantum counting. In *Proceedings of 25th ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 820–831. Springer, 1998. quant-ph/9805082.
9. D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society of London*, volume A439, pages 553–558, 1992.
10. E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation in determining parity. quant-ph/9802045, 16 Feb 1998.
11. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th STOC*, pages 212–219, 1996. quant-ph/9605043.
12. E. Hemaspaandra, L. A. Hemaspaandra, and M. Zimand. Almost-everywhere superiority for quantum polynomial time. quant-ph/9910033, 8 Oct 1999.
13. L. A. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986. Earlier version in STOC'84.
14. M. Mosca. Quantum searching, counting and amplitude amplification by eigenvector analysis. In *MFCS'98 workshop on Randomized Algorithms*, 1998.
15. A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of 31th STOC*, pages 384–393, 1999. quant-ph/9804066.
16. N. Nisan. CREW PRAMs and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991. Earlier version in STOC'89.
17. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94. quant-ph/9508027.
18. D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94.
19. J. S. Vitter and Ph. Flajolet. Average-case analysis of algorithms and data structures. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science. Volume A: Algorithms and Complexity*, pages 431–524. MIT Press, Cambridge, MA, 1990.
20. Ch. Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, 1999. quant-ph/9711070.