Table of Contents

Benedikt Löwe, Wolfgang Malzkorn, Thoralf Räsch
Foundations of the Formal Sciences II
Applications of Mathematical Logic in Philosophy and Linguistics
Bonn, November 10-13, 2000, pp. 1–18.

Quantum query algorithms and lower bounds

Andris Ambainis⋆
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540, USA
E-mail: `ambainis@cs.berkeley.edu`

Abstract. Many quantum algorithms can be analyzed in a query (oracle) model where input is given by a black box that answers queries and the complexity of the algorithm is measured by the number of queries to the black box that it uses. We survey the main results and techniques in this model, with an emphasis on lower bound methods.

1 Introduction

The modern version of the Church-Turing thesis asserts that any reasonable model of computation can be efficiently (with at most a polynomial slowdown) simulated by a probabilistic Turing machine. Quantum computing challenges this thesis from an unexpected direction.

The Turing machine model captures our intuition about how a general computation might proceed. Thus, either the modern Church-Turing thesis is correct or there is something missing in our intuition. The missing part is that our intuition is based on classical mechanics. Everything that we encounter in our everyday life and all of

Received: ...;
In revised version: ...;
Accepted by the editors: ....
2000 *Mathematics Subject Classification.* **PRIMARY** SECONDARY.
⋆ Supported by NSF Grant CCR-9987845 and the State of New Jersey.

today's computers function according to the principles of classical mechanics. Because of that, when we define an abstract model of computation like Turing machines, we are likely to incorporate some principles of classical mechanics into this model in an implicit form.

By using axioms of quantum mechanics instead, one can define quantum Turing machines and other models of quantum computation. There is evidence that those models are more powerful than their conventional (classical) counterparts. Simon [Sim97] showed an oracle problem which can be solved by a quantum algorithm in a polynomial time but requires exponential time classically. Shor [Sho97] gave a polynomial-time algorithm for factoring integers and discrete logarithms, problems that are conjectured to be very hard classically. Since public-key cryptosystems like RSA are based on the hardness of those problems, building a quantum computer would undercut the foundations of today's cryptography.

Another important quantum algorithm is due to Grover [Gro96]. It speeds up a very general class of search problems quadratically. While this speedup is just polynomial, Grover's algorithm applies to a large variety of problems which includes both NP-complete problems like satisfiability or 3-coloring and search problems solvable in polynomial time.

Almost all known quantum algorithms can be expressed in a query (oracle) model where the input is given by a black box which answers queries of a certain form. The algorithms that fall into this framework include Grover's algorithm, Simon's algorithm, period-finding (a key subroutine of Shor's factoring/discrete log algorithm) and many others. One of the advantages of this model is that we can also prove lower bounds on quantum algorithms.

In this paper, we describe the techniques for algorithms and lower bounds in the query model. We focus on Grover's search algorithm and present the algorithm and three methods of proving that this algorithm is optimal. Then, we list results for various other problems.

2 Definitions

2.1 Notation

We use $\oplus$ to denote XOR (exclusive OR). If $x_1 \in \{0,1\}$, $x_2 \in \{0,1\}$, $x_1 \oplus x_2$ denotes XOR of bits x_1, x_2. If $x_1 \in \{0,1\}^n$, $x_2 \in \{0,1\}^n$,

$x_1 \oplus x_2$ denotes bitwise XOR of n-bit strings x_1 and x_2. We use $+$ to denote the usual addition of integers.

We use the O and Θ notation [CorLeiRiv90] standard in computer science. Let $f(N)$ and $g(N)$ be functions defined on positive integers N and taking positive values. We say that $f = O(g)$ if there exists a constant c such that $f(N) \leq cg(N)$. We say that $f = o(g)$ if, for any $c > 0$, there exists N_0 such that $f(N) \leq cg(N)$ for all $N > N_0$. We say that $f = \Omega(g)$ if there exists $c > 0$ and N_0 such that $f(N) \geq cg(N)$ for all $N > N_0$.

2.2 Quantum computing

We introduce the basic model of quantum computing. For more details, see textbooks by Gruska [Gru99] and Nielsen and Chuang [NieChu00].

Quantum states: We consider finite dimensional quantum systems. An n-dimensional pure[1] quantum state is a vector $|\psi\rangle \in \mathbb{C}^n$ of norm 1. We use $|0\rangle$, $|1\rangle$, ..., $|n-1\rangle$ to denote an orthonormal basis for $\mathbb{C}^n$. Then, any state can be expressed as $|\psi\rangle = \sum_{i=0}^{n-1} a_i|i\rangle$ for some $a_0 \in \mathbb{C}$, $a_1 \in \mathbb{C}$, ..., $a_{n-1} \in \mathbb{C}$. Since the norm of $|\psi\rangle$ is 1, $\sum_{i=0}^{n-1} |a_i|^2 = 1$.

We call the states $|0\rangle$, ..., $|n-1\rangle$ *basis states*. Any state of the form $\sum_{i=0}^{n-1} a_i|i\rangle$ is called a *superposition* of $|0\rangle$, ..., $|n-1\rangle$. The coefficient a_i is called *amplitude* of $|i\rangle$.

A quantum system can undergo two basic operations: a unitary evolution and a measurement.

Unitary evolution : A *unitary transformation* U is a linear transformation on $\mathbb{C}^k$ that preserves the l_2 norm (i.e., maps vectors of unit norm to vectors of unit norm).

If, before applying U, the system was in a state $|\psi\rangle$, then the state after the transformation is $U|\psi\rangle$.

Measurements : In this survey, we just use the simplest case of quantum measurement. It is the full measurement in the computational basis. Performing this measurement on a state $|\psi\rangle = a_0|0\rangle + \ldots + a_{n-1}|n-1\rangle$ gives the outcome i with probability

[1] There are also mixed quantum states but they are not used in this paper. For the rest of the paper, "quantum state" means "pure quantum state".

$|a_i|^2$. The measurement changes the state of the system to $|i\rangle$. Notice that the measurement destroys the original state $|\psi\rangle$ and repeating the measurement gives the same i with probability 1 (because the state after the first measurement is $|i\rangle$).
More general classes of measurements are general von Neumann and POVM (Positive Operator-Valued Measure) measurements [NieChu00].

2.3 Query model

In the query model, the input $x_1, \ldots, x_N$ is contained in a black box and can be accessed by queries to the black box. In each query, we give i to the black box and the black box outputs x_i. The goal is to solve the problem with the minimum number of queries. The classical version of this model is known as *decision trees* [BuhdWo].

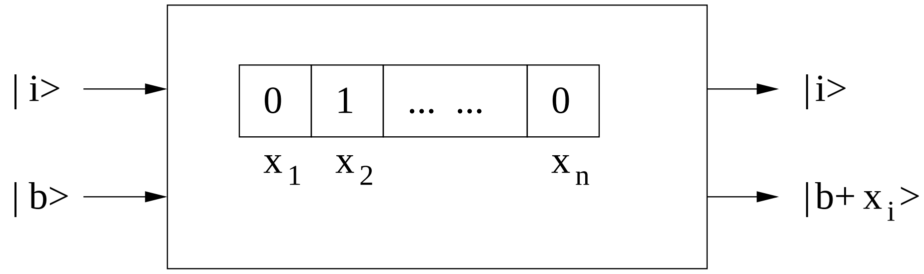

Fig. 1. *Quantum black box.*

There are two ways to define the query box in the quantum model. The first is an extension of the classical query (Figure 1). It has two inputs i, consisting of $\lceil \log N \rceil$ bits and b consisting of 1 bit. If the input to the query box is a basis state $|i\rangle|b\rangle$, the output is $|i\rangle|b \oplus x_i\rangle$. If the input is a superposition $\sum_{i,b} a_{i,b}|i\rangle|b\rangle$, the output is $\sum_{i,b} a_{i,b}|i\rangle|b \oplus x_i\rangle$. Notice that this definition applies both to the case when the x_i are binary and to the case when they are k-valued. In the k-valued case, we just make b consist of $\lceil \log_2 k \rceil$ bits and take $b \oplus x_i$ to be bitwise XOR of b and x_i.

In the second form of quantum query (which only applies to problems with $\{0,1\}$-valued x_i), the black box has just one input

i. If the input is a state $\sum_i a_i|i\rangle$, the output is $\sum_i a_i(-1)^{x_i}|i\rangle$. While this form is less intuitive, it is very convenient for use in quantum algorithms, including Grover's search algorithm [Gro96].

In this survey, we assume the first form as our main definition but use the second when describing Grover's algorithm. This is possible to do because a query of the second type can be simulated by a query of the first type [Gro96]. Conversely, an oracle of the first type can be simulated by a generalization of the sign oracle which receives $\sum_i a_{i,b}|i\rangle|b\rangle$ as an input and outputs $\sum_i a_i(-1)^{b\cdot x_i}|i\rangle|b\rangle$.

A quantum query algorithm with T queries is just a sequence of unitary transformations

$$U_0 \to O \to U_1 \to O \to \ldots \to U_{T-1} \to O \to U_T$$

on some finite-dimensional space $\mathbb{C}^k$. U_0, U_1, ..., U_T can be any unitary transformations that do not depend on the bits $x_1, \ldots, x_N$ inside the black box. O are query transformations that consist of applying the black box to the first $\log N + 1$ bits of the state. That is, we represent basis states of $\mathbb{C}^k$ as $|i, b, z\rangle$. Then, O maps $|i, b, z\rangle$ to $|i, b \oplus x_i, z\rangle$. We use O_x to denote the query transformation corresponding to an input $x = (x_1, \ldots, x_N)$.

The computation starts with the state $|0\rangle$. Then, we apply U_0, O_x, ..., O_x, U_T and measure the final state. The result of the computation is the rightmost bit of the state obtained by the measurement (or several bits if we are considering a problem where the answer has more than 2 values).

The quantum algorithm computes a function $f(x_1, \ldots, x_N)$ if, for every $x = (x_1, \ldots, x_N)$ for which f is defined, the probability that the rightmost bit of $U_T O_x U_{T-1} \ldots O_x U_0|0\rangle$ equals $f(x_1, \ldots, x_N)$ is at least $1 - \epsilon$ for some fixed $\epsilon < 1/2$.

The query complexity of f is the smallest number of queries used by a quantum algorithm that computes f. We denote it by $Q(f)$.

3 Grover's search algorithm

Consider the following problem.

Problem 1. Input: $x_1 \in \{0,1\}$, $x_2 \in \{0,1\}$, ..., $x_N \in \{0,1\}$ such that exactly one x_i is 1.

Output: the i such that $x_i = 1$.

Classically, one needs $\Omega(N)$ queries to solve this problem and there is no better algorithm than just querying the locations one by one until we find $x_i = 1$. Surprisingly, there is a better algorithm in the quantum case.

Theorem 1. *[Gro96] There is a quantum algorithm that solves Problem 1 with $O(\sqrt{N})$ queries.*

This is achieved by the following algorithm [Gro96].

1. Apply a unitary transformation U mapping $|0\rangle$ to $\frac{1}{\sqrt{N}}\sum_{i=1}^{N}|i\rangle$.
2. Repeat for $\lceil \frac{\pi}{4}\sqrt{N} \rceil$ times:
   (a) Apply the query transformation O which maps $\sum_{i=1}^{N} a_i|i\rangle$ to $\sum_{i=1}^{N} a_i(-1)^{x_i}|i\rangle$.
   (b) Apply the following "diffusion operator" D

$$\begin{cases} D|1\rangle = -\frac{N-2}{N}|1\rangle + \frac{2}{N}|2\rangle + \ldots + \frac{2}{N}|N\rangle \\ D|2\rangle = \frac{2}{N}|1\rangle - \frac{N-2}{N}|2\rangle + \ldots + \frac{2}{N}|N\rangle \\ \ldots \\ D|N\rangle = \frac{2}{N}|1\rangle + \frac{2}{N}|2\rangle + \ldots - \frac{N-2}{N}|N\rangle \end{cases}$$

3. Measure the state, output the result of the measurement.

We note that Grover's algorithm is efficient not just in the number of queries but also in the running time. The reason for that is that the diffusion operator D can be implemented in $O(\log N)$ time steps [Gro96]. Therefore, the whole algorithm can be implemented in $O(\sqrt{N}\log N)$ time.

We now informally explain why this algorithm works. Our presentation follows the "inversion against average" of Grover [Gro97]. A different visualization of Grover's algorithm (by two reflections of a vector) is described in [Aha99].

To understand the algorithm, plot the amplitudes of $|1\rangle$, ..., $|N\rangle$ at each step. After the first step, the state is $\frac{1}{\sqrt{N}}\sum_{i=1}^{N}|i\rangle$ and the amplitudes are $\frac{1}{\sqrt{N}}$ (Figure 2). After the first query, the amplitude of $|i\rangle$ with $x_i = 1$ becomes $-\frac{1}{\sqrt{N}}$ (Figure 3). Then, the diffusion operator D is applied.

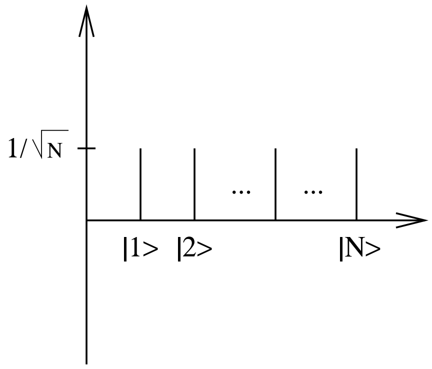

Fig. 2. Amplitudes before the first query

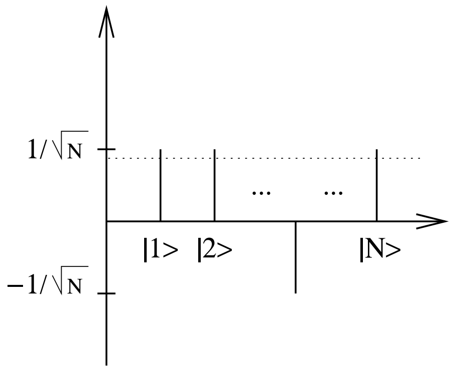

Fig. 3. Amplitudes after the first query

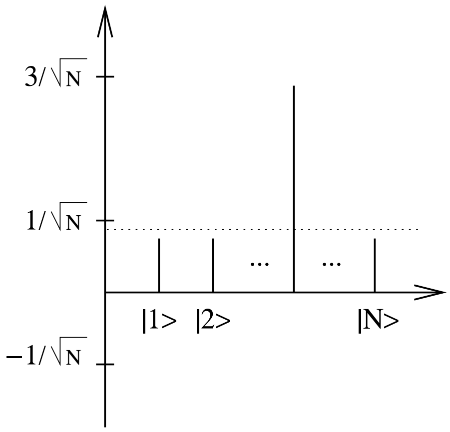

Fig. 4. Amplitudes after D

Let $|\psi\rangle = \sum_{i=1}^{N} a_i |i\rangle$ be the state before D. Then, the state after D is $|\psi'\rangle = \sum_{i=1}^{N} a'_i |i\rangle$ where $a'_i = -\frac{N-2}{N} a_i + \sum_{j \neq i} \frac{2}{N} a_j$. We can rewrite this as $a'_i = -a_i + \sum_{j=1}^{N} \frac{2}{N} a_j$ and $a'_i + a_i = \sum_{j=1}^{N} \frac{2}{N} a_j$. Let $A = \sum_{j=1}^{N} \frac{1}{N} a_j$ be the average of amplitudes a_i. Then, we have $a'_i + a_i = 2A$ and, if $a_i = A + \Delta$, then $a'_i = A - \Delta$. Thus, the effect of the "diffusion transform" is that every amplitude a_i is replaced by its reflection against the average of all a_i.

In particular, after the first query, the amplitude of $|i\rangle$ with $x_i = 1$ is $-\frac{1}{\sqrt{N}}$ and all the other amplitudes are $\frac{1}{\sqrt{N}}$. The average is $\frac{1}{\sqrt{N}} - \frac{2}{N\sqrt{N}}$ which is almost $\frac{1}{\sqrt{N}}$. Therefore, after applying D, the amplitude of $|i\rangle$ with $x_i = 1$ becomes almost $\frac{3}{\sqrt{N}}$ and the amplitudes of all other basis states $|j\rangle$ slightly less than $\frac{1}{\sqrt{N}}$ (Figure 4).

The next query makes the amplitude of $|i\rangle$ with $x_i = 1$ approximately $-\frac{3}{\sqrt{N}}$. The average of all amplitudes is slightly less than $\frac{1}{\sqrt{N}}$ and reflecting against it make the amplitude of $|i\rangle$ with $x_i = 1$ almost $\frac{5}{\sqrt{N}}$.

Thus, each step increases the amplitude of $|i\rangle$ with $x_i = 1$ by $O(1/\sqrt{N})$ and decreases the other amplitudes. A precise calculation [Gro96,Gro97] shows that, after $\frac{\pi}{4}\sqrt{N}$ steps, the amplitude of $|i\rangle$ with $x_i = 1$ is $1 - o(1)$. Therefore, the measurement gives the correct answer with probability $1 - o(1)$.

Boyer et.al. [BoyBraHoy98] have extended Grover's algorithm to the case when there can be more than one i with $x_i = 1$. The simplest case is if the number of $x_i = 1$ is known. If there are k such values, we can run the same algorithm with $\lceil \frac{\pi}{4}\sqrt{N/k} \rceil$ iterations instead of $\lceil \frac{\pi}{4}\sqrt{N} \rceil$. An analysis similar to one above shows that this gives a random i such that $x_i = 1$ with high probability.

A more difficult case is if k is not known in advance. The problem is that, after reaching the maximum, the amplitudes of $|i\rangle$ with $x_i = 1$ start to decrease. Therefore, if we do too many iterations, we might not get the right answer. This problem can be handled in two ways. The first is running the algorithm above several times with a different number of steps [BoyBraHoy98]. The second is invoking a different algorithm called "quantum counting" [BraHoyTap98] to estimate the number of $x_i = 1$ and then choose the number of steps for the search algorithm based on that. Either of those approaches gives us solution to:

Problem 2. **Input:** $x_1 \in \{0,1\}$, $x_2 \in \{0,1\}$, …, $x_N \in \{0,1\}$.
Output: i with $x_i = 1$, if there is one, "none" if $x_i = 0$ for all i.

Theorem 2. *[BoyBraHoy98] There is an algorithm that solves the problem 2 with $O(\sqrt{N})$ queries.*

Many problems can be solved by reductions to problems 1 and 2. For example, consider the satisfiability which is the canonical NP-complete problem. We have a Boolean formula $F(a_1, \ldots, a_n)$ and we have to find whether there exists a satisfying assignment ($a_1 \in \{0,1\}$, …, $a_n \in \{0,1\}$ for which $F(a_1, \ldots, a_n) = 1$). We can reduce the satisfiability to problem 2 by setting $N = 2^n$ and defining $x_1, \ldots, x_N$ to be $F(a_1, \ldots, a_n)$ for $N = 2^n$ possible assignments $a_1 \in \{0,1\}$, …, $a_n \in \{0,1\}$.

This means that we construct an algorithm that takes a_1, …, a_n and checks if $F(a_1, \ldots, a_n) = 1$. Then, if we replace the black box in the Grover's algorithm by this algorithm, we get an algorithm that finds a satisfying assignement in the time $O(\sqrt{2^n})$ times time needed to check one assignement.

A similar reduction applies to any other problem that can be solved by checking all possibilities in some search space.

4 Lower bounds for search

A very natural question is: can we do even better? Is there a more complicated algorithm that solves Problem 1 with even less queries? If this was the case, we would get better quantum algorithms for all the problems that reduce to it. In particular, an $O(\log^c N)$ query algorithm for Problem 1 would result in a polynomial-time quantum algorithm for satisfiability.

The next theorem shows that the speed-up achieved by Grover's algorithm is maximal.

Theorem 3. *[BenBerBra97] Any quantum algorithm for problem 2 uses $\Omega(\sqrt{N})$ queries.*

There are several proofs for this theorem using different techniques.

4.1 Classical adversary

Theorem 3 was first shown by Bennett et.al. [BenBerBra97] using a classical adversary method (also called "hybrid argument"). In this proof, we assume that we are given a quantum algorithm $\mathcal{A}$ that uses $o(\sqrt{N})$ queries. We then produce an input x_1, ..., x_N on which $\mathcal{A}$ outputs an incorrect answer.

To do that, we first analyze the algorithm on the input $x_1 = x_2 = \ldots = x_N = 0$ on which it must output "none". We find x_i such that the total probability of querying x_i at all steps is small. Then, we show that if we change x_i to 1 the final state of the algorithm does not change much. This implies that the algorithm either outputs the wrong answer on the original input $x_1 = x_2 = \ldots = x_N = 0$ or on the modified input $x_1 = \ldots = x_{i-1} = 0$, $x_i = 1$, $x_{i+1} = \ldots = x_N = 0$.

We now formalize this argument. Let $|\psi_i\rangle$ be the state before the i^{th} query (with the previous queries answered according to $x_1 = \ldots = x_N = 0$). We decompose this state as

$$|\psi_i\rangle = \sum_{j=1}^{N} \alpha_{ij} |j\rangle |\psi_{ij}\rangle$$

with $|j\rangle$ being the index that is queried and $|\psi_{ij}\rangle$ being the rest of the state. For each j, we consider the sum $S_i = \sum_i |\alpha_{ij}|$. This sum measures how much the algorithm queries the item x_j. Then, we have the following two lemmas.

Lemma 1. [BenBerBra97] Consider the algorithm $\mathcal{A}$ on inputs $x_1 = \ldots = x_N = 0$ and $x_1 = \ldots = x_{i-1} = 0$, $x_i = 1$, $x_{i+1} = \ldots = x_N = 0$. The probability distributions on possible answers on those inputs differ by at most $8S_i$.

Therefore, if the algorithm outputs "none" on the input $x_1 = \ldots = x_N = 0$ with probability at least $1 - \epsilon$, it also outputs "none" on $x_1 = \ldots = x_{i-1} = 0$, $x_i = 1$, $x_{i+1} = \ldots = x_N = 0$ with probability at least $1 - \epsilon - 8S_i$.

Lemma 2. [BenBerBra97] Let T be the number of queries. There exists $i \in \{1, \ldots, N\}$ with $S_i \leq \frac{T}{\sqrt{N}}$.

Therefore, if the number of queries is less than $c\sqrt{N}$ for a small constant c, the probability of the algorithm $\mathcal{A}$ outputting "none" on $x_1 = \ldots = x_{i-1} = 0$, $x_i = 1$, $x_{i+1} = \ldots = x_N = 0$ is close to $1 - \epsilon$ for some i. This means that $\mathcal{A}$ is incorrect on this input.

4.2 Polynomials method

The second approach [BeaBuhCle01] uses a completely different idea. We describe the probability of the algorithm outputting a given answer by a low-degree polynomial p. If the algorithm computes f with probability $1-\epsilon$, the polynomial approximates f within ϵ. Then, the lower bound on the number of queries follows from a lower bound on the degree of the polynomial approximating f. The key result is

Lemma 3. [BeaBuhCle01,ForRog99] Let $|\psi\rangle$ be the state of a quantum algorithm after k queries. Then, $|\psi\rangle = \sum_i \alpha_i(x_1, \ldots, x_N)|i\rangle$, with $\alpha_i(x_1, \ldots, x_N)$ being polynomials of degree at most k.

At the end of the algorithm, the probability of the measurement giving i is the square of the amplitude of $|i\rangle$. Since the amplitude is a polynomial of degree at most k, its square is a polynomial of degree at most $2k$.

Consider the version of problem 2 where the algorithm is only required to find if there is $x_i = 1$ or not. Let $p(x_1, \ldots, x_N)$ be the probability of the algorithm outputting the answer "there exists $x_i = 1$". Then, by the argument above, p is a polynomial in $x_1, \ldots, x_N$ of degree at most $2T$. If the algorithm is correct, this polynomial must satisfy the following properties:

1. $0 \leq p(0, \ldots, 0) \leq \epsilon$;
2. If $x_i = 1$ for some i, then $1 - \epsilon \leq p(x_1, \ldots, x_N) \leq 1$.

[NisSze92,Pat92] show that any polynomial satisfying these constraints has degree $\Omega(\sqrt{N})$. The first step of the proof is to use the symmetry to transform the polynomial in N variables into a polynomial in 1 variable $(x_1 + x_2 + \ldots + x_N)$ of the same degree. This gives a polynomial $p(x)$ such that $0 \leq p(0) \leq \epsilon$ but $1 - \epsilon \leq p(x) \leq 1$ for all $x \in \{1, \ldots, N\}$. Then, one can apply results from approximation theory [Che66] to show that any such polynomial must have degree $\Omega(\sqrt{N})$.

4.3 Quantum adversaries

The third approach is the "quantum adversary" method by Ambainis [Amb00]. The main idea is to run the algorithm on an input that is a quantum state. That is, we consider a quantum system consisting of two registers $\mathcal{H}_A$ and $\mathcal{H}_I$, with one register ($\mathcal{H}_A$) containing the algorithm's state and the other register ($\mathcal{H}_I$) containing the input. At the beginning $\mathcal{H}_A$ contains the algorithm's starting state ($|0\rangle$) and $\mathcal{H}_I$ contains a state of the form $\sum_x \alpha_x |x\rangle$ where $x = (x_1, x_2, \ldots, x_N)$ ranges over all possible inputs. If the algorithm performs a unitary transformation U_i, we just perform U_i on $\mathcal{H}_A$. Queries O are replaced by unitary transformations O' that are equal to O_x on the subspace consisting of states $|\psi\rangle|x\rangle$, $|\psi\rangle \in \mathcal{H}_A$.

If the input register $\mathcal{H}_I$ contains a basis state $|x\rangle$, then the state in $\mathcal{H}_A$ at any moment is just the state of the algorithm on the input x.

For example, consider the search (problem 1). First, we look at the case when $\mathcal{H}_I$ contains the basis state $|x\rangle$ with $x_i = 1$ and $x_j = 0$ for $j \neq i$. The correct answer on this input is i. Therefore, the end state in $\mathcal{H}_A$ must be $|i\rangle$. This means that the algorithm transforms the starting state containing $|0\rangle$ in $\mathcal{H}_A$ register into the end state containing $|i\rangle$ in $\mathcal{H}_A$:

$$|0\rangle|\underbrace{0\ldots0}_{i-1}1\underbrace{0\ldots0}_{N-i}\rangle \to |i\rangle|\underbrace{0\ldots0}_{i-1}1\underbrace{0\ldots0}_{N-i}\rangle. \tag{1}$$

We can also put a combination of several basis states $|x\rangle$ into $\mathcal{H}_I$. Then, we are running the algorithm on an input that is a quantum state. Since unitary transformations are linear, the end state of the algorithm on the starting state $|0\rangle \sum_x \alpha_x |x\rangle$ is just a linear combination of the end states for the starting states $|0\rangle|x\rangle$. For the search problem, we consider the starting state that is $\sum_x \frac{1}{\sqrt{N}}|x\rangle$, with summation over all x that have exactly one $x_i = 1$. Then, by summing up (1) for $i = 1, \ldots, N$, we get

$$\sum_{i=1}^{N} \frac{1}{\sqrt{N}}|0\rangle|\underbrace{0\ldots0}_{i-1}1\underbrace{0\ldots0}_{N-i}\rangle \to \sum_{i=1}^{N} \frac{1}{\sqrt{N}}|i\rangle|\underbrace{0\ldots0}_{i-1}1\underbrace{0\ldots0}_{N-i}\rangle. \tag{2}$$

Notice an important difference between the starting state and the end state in (2). In the starting state, the $\mathcal{H}_A$ and $\mathcal{H}_I$ registers are uncorrelated. For any state in $\mathcal{H}_I$, $\mathcal{H}_A$ register contains the same state $|0\rangle$. In the end state, the two registers are perfectly correlated. For every of N basis states $|0\ldots010\ldots0\rangle$ in $\mathcal{H}_I$, $\mathcal{H}_A$ contains a different state.

This is also true for other problems. In general, if the algorithm computes a function f, the parts of state in $\mathcal{H}_I$ that correspond to inputs x with one value of $f(x)$ become correlated with parts of $\mathcal{H}_A$ which correspond to the algorithm outputting this answer. (Those correlations are often called *quantum entanglement* because they are different from correlations between classical random variables [NieChu00].)

To bound the number of queries, we introduce a quantity measuring the correlations between $\mathcal{H}_A$ and $\mathcal{H}_I$. We show bounds on this quantity correlations in the starting state, end state and the maximum change in one query. This implies a lower bound on the number of queries. For more details, see [Amb00].

4.4 Comparison of different methods

While all three described methods give the same $\Omega(\sqrt{N})$ lower bound for problem 2, they are of different strength for other problems.

Classical adversary arguments can be usually recast into the quantum adversary framework. Thus, the quantum adversary method is a generalization of the classical adversary.

The strengths of the quantum adversary and the polynomials method are incomparable. For many problems, both of them give optimal lower bounds. The polynomials method usually needs the problem to have a certain degree of symmetry. The reason is that it is easier to deal with polynomials in 1 variable, rather than in N variables. For search and other symmetric functions, one can convert an N-variable polynomial to a 1 variable polynomial. However, if the problem lacks symmetry, it might be difficult.

On the other hand, the polynomials method is better for proving lower bounds for quantum algorithms with zero error or very small error [BuhCleWol99]. It is also the only method which gives a lower bound for the collision problem [Aar02,Shi01].

5 Other problems

A variety of other problems have been studied in the query model. We now list some main results and open problems.

5.1 Period-finding

This is a class of problems for which quantum algorithms have achieved the most impressive speedups. Two examples are:

Problem 3. [Sim97] We are given a black box computing a 2-1 function $f : \{0,1\}^n \to \{0,1\}^n$ such that there exists $x \in \{0,1\}^n$, $x \neq 0^n$ such that $f(x \otimes y) = f(y)$ for all y. We want to find x.

Classically, this problem requires querying $f(x)$ for $\Omega(2^{n/2})$ values of x. In the quantum case, $O(n)$ quantum queries are enough [Sim97]. This was the first example of an exponential advantage for a quantum algorithm. It led to Shor's factoring algorithm [Sho97] which is based on the following black-box problem.

Problem 4. [Sho97] We are given a black box computing a k-to-1 function $f : \mathbb{Z} \to \{1, \ldots, N\}$ such that $f(x) = f(x+k)$ for all x. We want to find k.

The problem can be solved with $O(1)$ queries [Sho97] (compared to $\Omega(\sqrt[3]{N})$ classically [Cle00]).

Both of those problems (and several other problems) are particular cases of the Abelian hidden subgroup problem [MosEke98].

5.2 Finding collisions

The algorithms for period-finding rely on the algebraic structure of the problem. If there is no such structure, the problem becomes much harder.

Problem 5. We are given $f : \{1, \ldots, N\} \to \{1, \ldots, N\}$ such that f is 2-1 and want to find x, y such that $f(x) = f(y)$.

Notice that problem 3 is a particular case of problem 5 with more structure ($f(x) = f(y)$ if and only if $y = x + z$). The absence of this structure changes the complexity dramatically. While problem 3 can be solved with $O(n) = O(\log N)$ queries, the best quantum algorithm for problem 5 uses $O(N^{1/3})$ queries [BraHoyTap97]. The algorithm uses Grover's search algorithm.

Proving a lower bound was open for 4 years, until Aaronson [Aar02] showed an $\Omega(N^{1/5})$ lower bound and Shi [Shi01] improved it to $\Omega(N^{1/3})$. The proof uses the polynomials method in a clever way. A related problem is element distinctness:

Problem 6. We are given $f : \{1, \ldots, N\} \to \{1, \ldots, N\}$ such that there is a unique x, y such that $x \neq y$ and $f(x) = f(y)$. We want to find x, y.

There is a quantum algorithm that solves element distinctness with $O(N^{3/4})$ queries [BuhDurHei01]. It is based on a recursive use of Grover's algorithm in a clever way. An $\Omega(N^{2/3})$ lower bound has been shown by [Shi01]. Closing the gap between those two results is an open problem. There is also a similar gap between upper and lower bounds for problems intermediate between 5 and 6 (functions f that have k pairs x, y such that $f(x) = f(y)$).

5.3 Counting and approximate counting

This is a class of problems studied by [Gro98,BraHoyTap98,NayWu99] and others. Up to quadratic speedup can be achieved. An example is

Problem 7. We are given $x_1 \in \{0,1\}$, $x_2 \in \{0,1\}$, ..., $x_N \in \{0,1\}$ and want to find $k = x_1 + \ldots + x_N$.

Classically, this requires $\Omega(N)$ queries. In the quantum case, $O(\sqrt{(k+1)(N-k+1)})$ queries are sufficient [BraHoyTap98]. If $k \approx N/2$, this is the same as $O(N)$ and there is no speedup. However, if k is close to 0 or N, quantum algorithms achieve up to quadratic speedup. (If $k = O(1)$, $O(\sqrt{(k+1)(N-k+1)}$ becomes $O(\sqrt{N})$ but classical algorithms still need $\Omega(N)$ queries.)

For $k \approx N/2$, although quantum algorithms need the same number of queries as classical for exact counting, they are better for approximate counting.

Problem 8 (ϵ-approximate count). We are given $x_1 \in \{0,1\}$, $x_2 \in \{0,1\}$, ..., $x_N \in \{0,1\}$ and we know that either at least $\frac{1}{2}+\epsilon$ fraction of them are 0 or at least $\frac{1}{2}+\epsilon$ fraction of them are 1. We want to distinguish these two cases.

This can be solved with quantum algorithms with $O(1/\epsilon)$ queries, compared with $O(1/\epsilon^2)$ queries needed classically [Gro98]. Both exact and approximate counting results are tight. Matching lower bounds can be shown by either the polynomials method [NayWu99] or the quantum adversary [Amb00].

For more information, see [BraHoyMos00,Gro98,NayWu99]

5.4 Binary search

Problem 9. We are given $x_1 = \ldots = x_i = 0$, $x_{i+1} = \ldots = x_N = 1$ and want to find i.

Classically, $\log_2 N$ queries are needed. Quantum algorithms can solve this problem with $0.53... \log_2 N$ [FarGolGut99] or $0.65... \log_2 N$ [HoyNeeShi01] queries using two different techniques. The best lower bound is $0.22... \log_2 N$ [HoyNeeShi01] by a variant of "quantum adversary". Thus, the speedup is at most a constant. It remains open to determine the exact constant achievable by quantum algorithms for this basic problem.

Related problems are searching and element distinctness, both of which have been studied in [HoyNeeShi01].

5.5 General bounds

There are classes of problems for which we can bound the maximum improvement over classical algorithms achievable for *any* problem in this class. The broadest such class is the class of total Boolean functions.

Let $f : \{0,1\}^N \rightarrow \{0,1\}$ be a *total* function of N variables $x_1 \in \{0,1\}$, ..., $x_N \in \{0,1\}$. Let $Q(f)$ be the quantum query complexity of f and $D(f)$ be the deterministic query complexity. Beals et al. [BeaBuhCle01] showed that $D(f) = O(Q^6(f))$. Thus, for total functions, quantum algorithms can have at most a polynomial speedup. This contrasts with partial functions for which an exponential speedup is possible. (An example is problem 3 or problem 4.)

It is open what is the maximum possible speedup achievable for total f. The best known example is the OR function

$$x_1\ OR\ x_2\ OR\ \ldots\ OR\ x_N.$$

Its deterministic query complexity is N but its quantum query complexity is $O(\sqrt{N})$ which is obtained by Grover's algorithm. (Solving problem 2 is enough to compute the OR function. If we find any $x_i = 1$, OR of all x_i is 1. If there is none, OR is 0.)

An excellent survey on this topic is [BuhdWo].

5.6 Other results

Van Dam [Dam98] has shown that any function of N 0,1-valued variables can be computed with $N/2 + o(N)$ queries. This is optimal because it is also known that any quantum algorithm needs $N/2$ queries to compute the parity of $x_1, \ldots, x_N$ [BeaBuhCle01,FarGolGut98].

Buhrman et.al. [BuhCleWol99] have studied the number of queries needed for zero-error quantum algorithms and small-error quantum algorithms. They showed that, for several problems, the speedup achieved by quantum algorithms disappears in the zero-error model. Thus, requiring a quantum algorithm to give the correct answer with probability $1-\epsilon$ (instead of 1) can be important for the construction of good quantum algorithms. They also studied the tradeoff between the decrease in the error and increase in the number of queries for the search problem.

Ambainis and de Wolf [AmbWol01] studied the average-case query complexity when we allow the algorithm to use different numbers of queries on different inputs and count the expected number of queries, under some distribution over possible values of $x_1, \ldots, x_N$.

6 Beyond the query model

A related area is quantum communication complexity. In the standard model of communication complexity [KusNis97], we have two parties (often called Alice and Bob) computing a function $f(x, y)$ of two inputs, with the input x known to one party (Alice) and the input y known to the other party (Bob). Since neither party knows both inputs, communication between the two parties is needed to compute f. The *communication complexity* of f is the minimum number of bits that need to be communicated between Alice and Bob to compute f.

The speedups for problems in the query model often imply similar speedups for communication problems. For example, the $O(\sqrt{N})$-query algorithm for problem 2 can be used to construct a communication protocol that solves the following problem (called *set disjointness*) by communicating just $O(\sqrt{N \log N})$ quantum bits[2] between Alice and Bob [BuhCleWig98]. This provides almost quadratic improvement over the classical case because, classically, $\Omega(N)$ bit communication is needed to solve this problem [KalSni92,Raz92].

Problem 10. Alice is given a subset $A \in \{1, \ldots, N\}$, Bob is given $B \in \{1, \ldots, N\}$. They want to find out if A and B are disjoint.

Query algorithms imply corresponding communication algorithms but proving lower bounds for quantum communication complexity is more difficult (because communication complexity is a more general setting). Proving a good lower bound for problem 10 was open for about 5 years until, very recently, Razborov [Raz02] showed that $\Omega(\sqrt{N})$ bits of communication are needed. The proof uses polynomials method in a combination with several other ideas.

For a survey on quantum communication complexity, see [Wol02]. A good book on classical communication complexity is [KusNis97].

Acknowledgements. Thanks to Yaoyun Shi and FOTFS referee for useful comments.

[2] The result has been further improved to $O(\sqrt{N} c^{\log^* n})$ bits by [HoyWol02].

References

[Aar02] S. Aaronson. Quantum lower bound for the collision problem. *Proceedings of STOC'02*, to appear.

[Aha99] D. Aharonov. Quantum computation. In *Annual Reviews of Computational Physics VI*, World Scientific, 1999.

[Amb00] A. Ambainis. Quantum lower bounds by quantum arguments. *Proceedings of STOC'00*, pp. 636-643. Journal version to appear in *Journal of Computer and System Sciences*.

[AmbWol01] A. Ambainis, R. de Wolf. Average-case quantum query complexity. *Journal of Physics A*, 34(2001), pages 6741-6754.

[BeaBuhCle01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(3):778-797, 2001.

[BenBerBra97] C. Bennett, E. Bernstein, G. Brassard, U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(3):1510-1523, 1997,

[BoyBraHoy98] M. Boyer, G. Brassard, P. Hoyer, A. Tapp. *Tight bounds on quantum searching. Fortschritte der Physik*, 46:493-505, 1998.

[BraHoyMos00] G. Brassard, P. Hoyer, M. Mosca and A. Tapp. Quantum amplitude amplification and estimation. *Quantum Information and Computation*, to appear. Also LANL preprint[3] quant-ph/0005055.

[BraHoyTap97] G. Brassard, P. Høyer, and A. Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News*, 28:14–19, 1997.

[BraHoyTap98] G. Brassard, P. Høyer, and A. Tapp. Quantum counting. *Proceedings of ICALP'98*, pp. 820-831.

[BuhCleWol99] H. Buhrman, R. Cleve, R. de Wolf, C. Zalka. Bounds for small-error and zero-error quantum algorithms. *Proceedings of FOCS'99*, pages 358-368.

[BuhCleWig98] H. Buhrman, R. Cleve, A. Wigderson. Quantum vs. classical communication and computation. *Proceedings of STOC'98*, pages 63-68.

[BuhDurHei01] H. Buhrman, C. Durr, M. Heiligman, P. Hoyer, F. Magniez, M. Santha, R. de Wolf. Quantum algorithms for element distinctness. *Proceedings of Complexity'01*, pp. 131-137.

[BuhdWo] H. Buhrman, R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, to appear.

[Che66] E. W. Cheney. *Introduction to Approximation Theory.* McGraw-Hill, 1966.

[Cle00] R. Cleve. The query complexity of order-finding. *Proceedings of Complexity'00*, pp. 54-59.

[CorLeiRiv90] T. Cormen, C. Leiserson, R. Rivest. *Introduction to Algorithms.* MIT Press, 1990

[Dam98] W. van Dam. Quantum Oracle Interrogation: Getting all information for almost half the price. *Proceedings of FOCS'98*, pp. 362-367.

[Deu85] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London, Series A*, 400:97-117, 1985.

[FarGolGut98] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, A limit on the speed of quantum computation in determining parity, *Phys. Rev. Lett.* 81:5442-5444, 1998.

[3] quant-ph preprints are available at `http://www.arxiv.org/archive/quant-ph`

[FarGolGut99] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Invariant Quantum Algorithms for Insertion into an Ordered List. quant-ph/9901059.

[ForRog99] L. Fortnow, J. Rogers. Complexity limitations on quantum computing. *Journal of Computer and System Sciences*, 59 : 240-252, 1999. Also Complexity'98.

[Gro96] L. Grover. A fast quantum mechanical algorithm for database search, *Proceedings of STOC'96*, pp. 212-219

[Gro97] L. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79:325-328, 1997.

[Gro98] L. Grover. A framework for fast quantum mechanical algorithms. *Proceedings of STOC'98,* pp. 53-62.

[Gru99] J. Gruska. *Quantum Computing.* McGraw-Hill, 1999.

[HoyNeeShi01] P. Hoyer, J. Neerbek, Y.Shi. Quantum bounds for ordered searching and sorting. *Proceedings of ICALP'01*, pp. 346-357.

[HoyWol02] P. Hoyer, R. de Wolf. Improved quantum communication complexity bounds for disjointness and equality. *Proceedings of STACS'02*, pp. 299-310

[KalSni92] B. Kalyanasundaram, G. Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5:545–557, 1992.

[KusNis97] E. Kushilevitz, N. Nisan. *Communication Complexity.* Cambridge University Press, Cambridge, 1997.

[MosEke98] M. Mosca, A. Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. *Proceedings of QCQC'98*, pp. 174-188.

[NayWu99] A. Nayak, F. Wu. The quantum query complexity of approximating the median and related statistics. *Proceedings of STOC'99*, pp. 384–393.

[NisSze92] N. Nisan, M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4:301-313, 1994. Also *STOC'92*.

[NieChu00] M. Nielsen, I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[Pat92] R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions. *Proceedings of STOC'92*, pp. 468-474.

[Raz92] A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106:385-390, 1992.

[Raz02] A. Razborov. Quantum communication complexity of symmetric predicates, quant-ph/0204025.

[Shi01] Y. Shi, Quantum lower bounds for the collision and the element distinctness problems, quant-ph/0112086.

[Sho97] P. Shor. Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing.* 26:1484-1509, 1997.

[Sim97] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26:1474-1483, 1997.

[Vaz98] U. Vazirani. On the power of quantum computation. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical and Physical Sciences*, 356:1759-1768, August 1998.

[Wol02] R. de Wolf. Quantum communication and complexity. *Theoretical Computer Science*, to appear.