A Limit on the Speed of Quantum Computation in Determining Parity

Edward Farhi and Jeffrey Goldstone
Center for Theoretical Physics
Massachusetts Institute of Technology
Cambridge, MA 02139

Sam Gutmann
Department of Mathematics
Northeastern University
Boston, MA 02115

Michael Sipser
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139

Consider a function $f$ which is defined on the integers from 1 to $N$ and takes the values $-1$ and $+1$. The parity of $f$ is the product over all $x$ from 1 to $N$ of $f(x)$. With no further information about $f$, to classically determine the parity of $f$ requires $N$ calls of the function $f$. We show that any quantum algorithm capable of determining the parity of $f$ contains at least $N/2$ applications of the unitary operator which evaluates $f$. Thus for this problem, quantum computers cannot outperform classical computers.

I. INTRODUCTION

If a quantum computer is ever built, it could be used to solve certain problems in less time than a classical computer. Simon found a problem that can be solved exponentially faster by a quantum computer than by the provably best classical algorithm \[1\]. The Shor algorithm for factoring on a quantum computer gives an exponential speedup over the best known classical algorithm \[2\]. The Grover algorithm gives a speedup for the following problem \[3\]. Suppose you are given a function $f(x)$ with $x$ an integer and $1 \leq x \leq N$. Furthermore you know that $f$ is either identically equal to 1 or it is 1 for $N-1$ of the $x$’s and equal to $-1$ at one unknown value of $x$. The task is to determine which type of $f$ you have. Without any additional information about $f$, classically this takes of order $N$ calls of $f$ whereas the quantum algorithm runs in time of order $\sqrt{N}$. In fact this $\sqrt{N}$ speedup can be shown to be optimal \[4\].

It is of great interest to understand the circumstances under which quantum speedup is possible. Recently Ozhigov has shown that there is a situation where a quantum computer cannot outperform a classical computer \[5\]. Consider a function $g(t)$, defined on the integers from 1 to $L$, which takes integer values from 1 to $L$. We wish to find the $M$th iterate of some input, say 1, that is, $g^M(1)$. (Here $g^{(n)}(t) = g(g^{(n-1)}(t))$ and $g^{(0)}(t) = t$.) Ozhigov’s result is that if $L$ grows at least as fast as $M^6$ then any quantum algorithm for evaluating the $M$th iterate takes of order $M$ calls of the unitary operator which evaluates $g$; of course the classical algorithm requires $M$ calls. Later we will show that our result in fact implies a stronger version of Ozhigov’s with $L = 2M$.

In this paper we show that a quantum computer cannot outperform a classical computer in determining the parity of a function; similar and additional results are obtained in \[6\] and \[7\]. Let

$$f(x) = \pm 1 \quad \text{for} \quad x = 1, \ldots, N .$$

Define the parity of $f$ by

$$\text{par}(f) = \prod_{x=1}^{N} f(x)$$

so that the parity of $f$ can be either $+1$ or $-1$. The parity of $f$ always depends on the value of $f$ at every point in its domain so classically it requires $N$ function calls to determine the parity. The Grover problem, as described above,
is a special case of the parity problem where additional restrictions have been placed on the function. Although the Grover problem can be solved in time of order $\sqrt{N}$ on a quantum computer, the parity problem has no comparable quantum speedup.

II. PRELIMINARIES

We imagine that the function $f$ whose parity we wish to determine is provided to us in the form of an ordinary computer program, thought of as an oracle. We then use a quantum compiler to convert this to quantum code which gives us the unitary operator

$$U_f|x,+1\rangle = |x,f(x)\rangle$$
$$U_f|x,-1\rangle = |x,-f(x)\rangle .$$

(Here the second register is a qubit taking the values $\pm 1$.) Defining

$$|x,s\rangle = \frac{1}{\sqrt{2}}(|x,+1\rangle + |x,-1\rangle)$$

and

$$|x,a\rangle = \frac{1}{\sqrt{2}}(|x,+1\rangle - |x,-1\rangle) ,$$

we have that

$$U_f|x,q\rangle = f(x,q)|x,q\rangle \quad q = s,a$$

where

$$f(x,s) = 1 \quad \text{and} \quad f(x,a) = f(x) .$$

Therefore in the $|x,q\rangle$ basis, the quantum operator $U_f$ is multiplication by $f(x,q)$.

Suppose that $N = 2$ so that $x$ takes only the values 1 and 2. Then

$$U_f(|1,a\rangle + |2,a\rangle) = f(1)|1,a\rangle + f(2)|2,a\rangle$$
$$= f(1)(|1,a\rangle + \text{par}(f)|2,a\rangle) .$$

Now the states $|1,a\rangle + |2,a\rangle$ and $|1,a\rangle - |2,a\rangle$ are orthogonal so we see that one application of $U_f$ determines the parity of $f$ although classically two function calls are required. See for example [8]. In section IV this algorithm is generalized for the case of $N$ to determine parity after $N/2$ applications of $U_f$.

In writing (3) we ignored the work bits used in calculating $f(x)$. This is because, quite generally, the work bits can be reset to their $x$ independent values [9]. To do this you must first copy $f(x)$ and then run the quantum algorithm for evaluating $f(x)$ backwards thereby resetting the work bits. If this is done then a single application of $U_f$ can be counted as two calls of $f$.

III. MAIN RESULT

We imagine that we have a quantum algorithm for determining the parity of a function $f$. The Hilbert space we are working in may be much larger than the $2N$-dimensional space spanned by the vectors $|x,q\rangle$ previously described. The algorithm is a sequence of unitary operators which acts on an initial vector $|\psi_0\rangle$ and produces $|\psi_f\rangle$. The Hilbert space is divided into two orthogonal subspaces by a projection operator $\mathcal{P}$. After producing $|\psi_f\rangle$, we measure $\mathcal{P}$ obtaining either 0, corresponding to parity $-1$, or 1, corresponding to parity $+1$. (Note that $\langle \psi_f | \mathcal{P} | \psi_f \rangle$ is the probability of obtaining 1.) We say that the algorithm is successful if there is an $\epsilon > 0$ such that

For $\text{par}(f) = +1$, \hspace{1cm} \langle \psi_f | \mathcal{P} | \psi_f \rangle \geq \frac{1}{2} + \epsilon$

and

For $\text{par}(g) = -1$, \hspace{1cm} \langle \psi_g | \mathcal{P} | \psi_g \rangle \leq \frac{1}{2} - \epsilon .

(8)
This is a weak definition of success for an algorithm—we only ask that the probability of correctly identifying the parity of $f$ be greater than $\frac{1}{2}$ no matter what $f$ is. Since we are proving the nonexistence of a successful (short) algorithm, our result is correspondingly strong.

The algorithm is a sequence of unitary operators, some of which are independent of $f$, and some of which depend on $f$ through the application of a generalization of (8). We need to generalize (8) because we are working in a larger Hilbert space. In this larger Hilbert space there are still subspaces associated with $|a,q,w\rangle$ where $x = 1,\ldots,N$ and $q = a, s$ and $w = 1,\ldots,W$ for some $W$, corresponding to the values of the work bits that the algorithm may use.) Accordingly there are projection operators $P_x$ and $P_q$ which obey

$$P^2_x = P_x; \quad P_x P_y = 0 \text{ for } x \neq y; \quad \sum_{x=1}^{N} P_x = 1$$

and

$$P^2_q = P_q; \quad P_s P_a = 0; \quad \sum_{q=s,a} P_q = 1. \quad (9)$$

In terms of these projectors we have

$$U_f = \sum_x \sum_q f(x,q)P_x P_q \quad (10)$$

where the sum over $x$ is from 1 to $N$ and the sum over $q$ is over $s$ and $a$.

An algorithm which contains $k$ applications of $U_f$, acting on $|\psi_0\rangle$, produces

$$|\psi_f\rangle = V_k U_f V_{k-1} U_f \ldots V_1 U_f |\psi_0\rangle \quad (11)$$

where $V_1$ through $V_k$ are unitary operators independent of $f$, but which may involve the work bits. For more extensive discussion, see [10].

We will now use (10) to put $\langle \psi_f|P|\psi_f\rangle$ in a form where we can see explicitly how it depends on $f$, allowing us to show that (8) is impossible if $k$ is too small. We have

$$\langle \psi_f|P|\psi_f\rangle = \sum_{x_1,q_1} \sum_{x_2,q_2} \ldots \sum_{x_{2k},q_{2k}} A(x_1,q_1 \ldots x_{2k},q_{2k}) \prod_{i=1}^{2k} f(x_i,q_i) \quad (12)$$

where

$$A(x_1,q_1 \ldots x_{2k},q_{2k}) = \langle \psi_0|P_{x_1} P_{q_1} V_1^{\dagger} \ldots V_{k}^{\dagger} V_{k} \ldots V_1 P_{x_{2k}} P_{q_{2k}} |\psi_0\rangle. \quad (13)$$

Note that $A$ does not depend on $f$.

There are $2^N$ different possible $f$’s of the form given by (8). We now sum over all these functions and compute

$$\sum_f \langle \psi_f|P|\psi_f\rangle \text{par}(f) = \sum_f \sum_{x_1,q_1} \sum_{x_{2k},q_{2k}} A(x_1,q_1 \ldots x_{2k},q_{2k}) \prod_{i=1}^{2k} f(x_i,q_i) \prod_{y=1}^{N} f(y). \quad (14)$$

Note that

$$\sum_f f(z) = 0 \text{ for } z = 1,\ldots,N \quad (15)$$

because for each function with $f(z) = +1$ there is a function with $f(z) = -1$. Similarly if $z_1, z_2 \ldots z_n$ are all distinct, we have

$$\sum_f f(z_1)f(z_2)\ldots f(z_n) = 0. \quad (16)$$

Return to (14) and consider the sum on $f$,
\[
\sum_{f} \prod_{i=1}^{2k} f(x_i, q_i) \prod_{y=1}^{N} f(y)
\] (17)

where \(x_1, x_2 \ldots x_{2k}\) and \(q_1, q_2 \ldots q_{2k}\) are fixed. For any \(i\) with \(q_i = s\) we have \(f(x_i, s) = 1\). Thus (17) equals

\[
\sum_{f} \prod_{i=1}^{2k} f(x_i) \prod_{y=1}^{N} f(y)
\] (18)

Now \(f(z) = 1\) for any \(z\) and any \(f\). By (16), the sum over \(f\) in (18) will give 0 unless each term in the second product can be matched to a term in the first product. Since the first product has at most \(2k\) terms and the second product has \(N\) terms, we see that if \(2k < N\) then the sum over \(f\) in (18) is 0 and accordingly,

\[
\sum_{f} \langle \psi_f | P | \psi_f \rangle \text{par}(f) = 0
\] (19)

This implies that for \(2k < N\)

\[
\sum_{f, \text{par}(f) = +1} \langle \psi_f | P | \psi_f \rangle = \sum_{f, \text{par}(f) = -1} \langle \psi_f | P | \psi_f \rangle
\] (20)

which means that for \(k < N/2\) condition (8) cannot be fulfilled.

Equation (20) shows that our bound holds even if we further relax the success criterion given in condition (8). In any algorithm with fewer than \(N/2\) applications of \(U_f\), demanding a probability of success greater than or equal to \(1/2\) for every \(f\) forces the probability to be \(1/2\) for every \(f\).

### IV. AN OPTIMAL ALGORITHM

To see that the bound \(k < N/2\) is optimal, we now show how to solve the parity problem with \(N/2\) applications of \(U_f\). Here we assume that \(N\) is even. We only need the states \(|x,a\rangle\) given in (4) for which \(U_f |x,a\rangle = f(x) |x,a\rangle\).

Define

\[
V|x,a\rangle = |x+1,a\rangle \quad x = 1, \ldots, \frac{N}{2} - 1
\]

\[
V|\frac{N}{2},a\rangle = |1,a\rangle
\]

\[
V|x,a\rangle = |x+1,a\rangle \quad x = \frac{N}{2} + 1, \ldots, N - 1
\]

\[
V|N,a\rangle = |\frac{N}{2} + 1,a\rangle
\] (22)

Also let

\[
|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x,a\rangle
\] (23)

Now compute \(|\psi_f\rangle\) given by (11) with \(k = N/2\) and for the operators independent of \(f\) take

\[V_1 = V_2 = \ldots = V_{k-1} = V \quad \text{and} \quad V_k = 1.\]

We then have that

\[
|\psi_f\rangle = \frac{1}{\sqrt{N}} f(1)f(2) \ldots f(\frac{N}{2}) \sum_{x=1}^{N/2} |x,a\rangle + \frac{1}{\sqrt{N}} f(\frac{N}{2} + 1)f(\frac{N}{2} + 2) \ldots f(N) \sum_{x=\frac{N}{2} + 1}^{N} |x,a\rangle
\] (24)

Therefore if \(\text{par}(f) = +1\), the state \(|\psi_f\rangle\) is proportional to \(|\psi_0\rangle\) whereas if \(\text{par}(f) = -1\), then \(|\psi_f\rangle\) is orthogonal to \(|\psi_0\rangle\). For the parity projection operator we take \(P = |\psi_0\rangle \langle \psi_0|\) and we see that the algorithm determines the correct parity all the time. Similarly we can show that if \(N\) is odd, then with \(k = (N + 1)/2\) applications of \(U_f\) we can determine the parity of \(f\), but this time we need the states \(|x,s\rangle\) as well as \(|x,a\rangle\).
Here we are interested in evaluating the $N$th iterate of a function which maps a set of size $2N$ to itself. We show that it is impossible for a quantum computer to solve this problem with fewer than $N/2$ applications of the unitary operator corresponding to the function. As noted above, this is a considerable strengthening of Ozhigov’s result.

We assume an algorithm satisfying the above conditions exists and we obtain a contradiction. Let the set of $2N$ elements be $\{(x,r)\}$ where $x = 1, \ldots, N$ and $r = \pm 1$. For any $f$ of the form (1) define

$$g(x, r) = (x + 1, rf(x))$$

(25)

where we interpret $N + 1$ as 1. Note that

$$g^{[N]}(1, 1) = (1, \text{par}(f))$$

(26)

Thus an algorithm which computes the $N$th iterate of $g$ with fewer than $N/2$ applications of the corresponding unitary operator would in fact solve the parity problem impossibly fast.

VI. CONCLUSION

Grover’s result raised the possibility that any problem involving a function with $N$ inputs could be solved quantum mechanically with only $\sqrt{N}$ applications of the corresponding operator. We have shown that this is not the case. For the parity problem, $N/2$ applications of the quantum operator are required.

Acknowledgment

Three of us are grateful to the fourth.