Quantum advantage and noise reduction in distributed quantum computing

J. Avron, Ofer Casper and Ilan Rozen

Department of Physics, Technion, 320000 Haifa, Israel

August 23, 2021

Abstract

Distributed quantum computing can give substantial noise reduction due to shallower circuits. An experiment illustrates the advantages in the case of Grover search. This motivates studying the quantum advantage of the distributed version of the Simon and Deutsch-Jozsa algorithm. We show that the distributed Simon algorithm retains the exponential advantage, but the complexity deteriorates from $O(n)$ to $O(n^2)$, where $n = \log_2(N)$. The distributed Deutsch-Jozsa deteriorates to being probabilistic but retains a quantum advantage over classical random sampling.

1 Introduction

The IBM quantum experience and Amazon Braket offer the opportunity to implement quantum algorithms on many small and noisy quantum computers. More than 20 quantum computers, with at most 65 qubits have been deployed by IBM. None can communicate quantumly. The question then begs itself what advantages and disadvantages distributed quantum computing with classical communication offer.

Replacing quantum by classical resources usually leads to large overhead. For example, simulating $n$ qubits needs $O(N = 2^n)$ classical bits. More generally, simulating a quantum circuit with $n+k$ qubits by a quantum circuits with $n$ qubits requires\footnote{c is a function of $n$.} $O(2^{ck})$ uses of the quantum circuits \cite{5}.

How much quantum advantage survives in distributed computing depends on the algorithms. Cirac et. al. \cite{6} showed that distributed 3SAT retains quantum advantage. Bravyi et. al \cite{5} estimated the overhead in classical computation for sparse quantum circuits and Peng et. al. \cite{12} derived related results for tensor networks with limited connections between clusters.

Distributed quantum computing can offer, besides the obvious advantage of additional “virtual qubits”, the advantage of significant noise reduction. This comes about because splitting an algorithm can result in significant reduction of depth. Since the noise in the output scales exponentially with the depth of the circuit this can be a significant advantage. For example, if the depth of a circuit is large enough the output of the quantum computer may be overwhelmed by noise, but a distributed computation with shallower depth may give significant results. As far as we an tell, this simple, one may say trivial, point has not been studied before.
We shall describe an experiment involving Grover search\cite{grover1996fast} that illustrate the advantage of distributed quantum computing over the undistributed computation. In addition we study the quantum advantage of distributed algorithms, for two basic textbook examples: Simon’s\cite{simon1997rapid,venegas2006quantum} and Deutsch-Jozsa\cite{deutsch1985rapid}. Since the Grover and Deutsch-Jozsa algorithm involve the use of an Oracle we shall also consider the task of distributing an Oracle.

As we shall see:

- The distributed Simon’s algorithm retains the exponential speedup albeit with higher complexity, see section 5.1.
- The quantum advantage of the distributed Deutsch-Jozsa deteriorates dramatically, see section 6.

2 Distributed computations of Boolean functions

We restrict ourselves to a particular scheme of distributed computing which suffices to cover the problems we consider.

2.1 Distributed classical computation

Consider a (classical) circuit with $n$ bits that computes the function

$$f : \{0,1\}^n \rightarrow \{0,1\}^m, \quad m \leq n - 1 \quad (2.1)$$

The function can be split into its even and odd parts: $f_{\text{even/odd}} : \{0,1\}^{n-1} \rightarrow \{0,1\}^m$

$$f_{\text{even}}(y_1, \ldots, y_m) = f(y_1, \ldots, y_m, 0), \quad f_{\text{odd}}(y_1, \ldots, y_m) = f(y_1, \ldots, y_m, 1) \quad (2.2)$$

More generally,

$$f_{\text{even}}(y_1, \ldots, y_m) = f(y_1, \ldots, y_{j-1}, 0, y_{j+1}, \ldots, y_{m+1}),$$

$$f_{\text{odd}}(y_1, \ldots, y_m) = f(y_1, \ldots, y_{j-1}, 1, y_{j+1}, \ldots, y_{m+1}) \quad (2.3)$$

We assume that the even and odd parts can be computed by a circuit with $n - 1$ bits. We can then distribute computing $f$ to two $n - 1$ bits processors.

2.2 Distributed quantum computation

Suppose Alice has a processor with $n$ connected qubits and Bob has two devices with $n - 1$ connected qubits each. A general state of Alice’s processor

$$|\psi\rangle = \sum_{j=0}^{N-1} \psi_j \, |j\rangle, \quad N = 2^n \quad (2.4)$$

is described by $N$ amplitudes. The state of Bob’s two processors is

$$|\phi_1\rangle \otimes |\phi_2\rangle, \quad |\phi_j\rangle = \sum_{k=0}^{N/2-1} \phi_{jk} \, |k\rangle \quad (2.5)$$

and it, too, is described by $2N/2 = N$ amplitudes.
Alice’s quantum advantage comes from her ability to entangle her \( n \) qubits. Bob can not entangle the two processors and can only entangle \( n-1 \) qubits on each processor. Bob has more qubits and shallower circuits. This is an advantage in noisy quantum computers. Bob also has the advantage that his measurements give \( 2(n-1) \) bits of information while Alice’s measurement gives her only \( n \) bits of information. We shall, however, not make use of this advantage here.

### 2.3 Quantum circuits and DNF

Grover and Deutsch-Jozsa algorithms involve the use of Oracles. We therefore face the problem of distributing an Oracle without introducing bias. To do so we assume that the Oracle is an algorithm with a standard form\(^2\) that allows to split it to the even and odd arguments. The reader who is willing to accept this on good faith may want to skip to the next section.

The Oracle, by definition, computes a Boolean function. Any Boolean function can be represented by DNF. (For an elementary introduction see appendix A). For example, the DNF of the function that assigns 1 to 101 and 010 and 0 otherwise is

\[
f(x_0, x_1, x_2) = x_0 \cdot \bar{x}_1 \cdot x_2 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2
\]

where \( x_j \in \{0, 1\} \) are logical variables (equivalently, binaries) and \( \bar{x}_j \) is the (logical) NOT, (equivalently, for binary \( \bar{x}_j = x_j \oplus 1 \)). The \(+\) is the (logical) OR normally denoted \( \lor \).

In the general case with \( n \) logical variables \( x_j, j \in 1, \ldots, n \), the DNF could be a sum of a large number of terms, each term is a product over all logical variables; Each variable appears once either as \( x_j \) or as \( \bar{x}_j \).

The quantum circuit that computes

\[
U_f |x\rangle = (-1)^{f(x)} |x\rangle
\]

can be read out from the DNF. For the DNF in Eq. (2.6) the circuit is given in Fig. 1. In the general case, a pair of \( X \) gates decorate all the \( \bar{x} \) and the \( n \)-fold product is represented by \( C^n Z \).

**Remark 2.1.** The DNF is, in general, not the most compact representation of the function \( f \) and similarly, the corresponding quantum circuit need not be the optimal circuit. Optimization of quantum circuits is considered in [2].

The even and odd parts of \( f \) are easily constructed from the DNF. If we use \( x_0 \) as the bit that determine even/odd then \( f_{e/o} \) for the example in Eq. (2.6) is

\[
f_e = x_1 \cdot \bar{x}_2, \quad f_o = \bar{x}_1 \cdot x_2
\]

---

\(^2\)In computer science one normally does not worry how the Oracles does what it does. But, for the case at hand, we need to.
In the general case, all the $\bar{x}_0$ terms make the even part (with $\bar{x}_0$ deleted) all the $x_0$ terms make the odd part (with $x_0$ deleted). The DNF is then used to construct the $n - 1$ qubits circuits for the even and odd parts:

$$ U_0^n : X \rightarrow X, \quad U_1^n : X \rightarrow X $$

Figure 2: The circuits corresponding to the DNF of Eq. (2.8)

In conclusion, given Alice’s $n$ qubit circuit corresponding to the DNF, there is a simple procedure that constructs the $n - 1$ qubits circuits of Bob for the even and odd parts of the function.

An algorithm for generating the even and odd circuits is:

**Algorithm 2.1: Splitting a Circuit**

Result: two (n-1)-qubit circuits $U_{fn}^e, U_{fn}^o$

1. Divide ($U_{fa}$)
   
2. $parity \leftarrow 0$;
3. $U_{fn}^o \leftarrow I$;
4. $U_{fn}^e \leftarrow I$;
5. if $U_{fa}$ is not of DNF then
   
6. abort ;
7. foreach gate $G$ in $U_{fa}$ in the order they are executed from top to bottom do
   
8. if $G$ is an X gate acting on the parity qubit $a$ then
   
9. $parity \leftarrow NOT(parity)$ ;
10. else if $G$ is a $C^{(n-1)}Z$ then
     
11. if $parity == 1$ then
     
12. append $C^{(n-2)}Z$ to $U_{fn}^e$;
13. else
     
14. append $C^{(n-2)}Z$ to $U_{fn}^o$;
15. else if $G$ is a single qubit gate acting not on the parity qubit then
     
16. append $G$ to its respective qubit both in $U_{fn}^o$ and $U_{fn}^e$ ;
17. else
     
18. abort;
19. return $U_{fn}^o, U_{fn}^e$ ;

---

*aThe parity qubit is defined as the qubit that distinguishes between the even and odd subspaces*

The algorithm returns the requisite $U_{fn}^o$ and $U_{fn}^e$:

$$ U_{fa} |1\rangle \otimes |\varphi\rangle \equiv |1\rangle \otimes U_{fn}^o |\varphi\rangle \quad U_{fa} |0\rangle \otimes |\varphi\rangle \equiv |0\rangle \otimes U_{fn}^e |\varphi\rangle $$

(2.9)

In appendix B we shall describe the converse procedure whereby starting from Bob’s distributed quantum circuits, Alice can generate her single quantum circuit.
3 The depth of distributed computation

Distributed algorithms have an advantage of shallower depth. This is evident for circuits given by DNF: The depth of the $n$ qubits circuit is distributed between the two $n - 1$ circuits. (See Eqs. (2.6,2.8).)

In the experiment, section 4, the depth of the distributed circuits was about a factor 4 smaller than the undistributed circuit. This is a consequence of the fact that the depth of a $C^n Z$ circuit is considerably larger than the depth of $C^{n-1} Z$. We illustrate this with an example.

The gain in depth depends on the choice of gates. We use the following rules [3]:

- Any single qubit gate is allowed
- The only 2-qubit gate allowed is CNOT
- Gates that can be executed in parallel are grouped into columns
- Consecutive single qubits gates are counted as a single qubit gate

Consider the circuit in Fig. 1 and its distributed cousins Fig. 2. To compute $\text{depth}(U_{f_A})$ and $\text{depth}(U_{f_B})$. We first observe that

$$\text{depth}(C^2 Z) = 11 \quad (3.1)$$

This is seen from Fig. 3

It follows that

$$\text{depth}(U_{f_A}) = 3 + 2 \cdot \text{depth}(C^2 Z) = 25 \quad (3.2)$$

To compute the depth of the corresponding distributed circuits in Fig. 2 we first recall

$$\equiv \quad (3.3)$$

For the even circuit one combines the $X$ gates with the Hadamard gates. Hence both circuits have:

$$\text{depth}(U_{f_n}^{e/o}) = 3$$

The example shows that distributed circuits can lead to substantial reduction in the depth of quantum circuits.
4 Distributed Grover: An experiment

In this section we describe an experiment carried on the IBMQ5 (ibmq_santiago). The data for the machine (at the time of the experiments) are given in Table 1.

In the experiment a Grover search for a single target among 16 items, \((N = 16\) and \(M = 1)\) was conducted on an undistributed circuit with \(n = 4\) qubits and then on distributed circuits with \(n = 3\) qubits each. The target state in both cases has been \(|1111\rangle\).

We made use of the open-source Qiskit library to generate the Grover search circuits which were then run on real (i.e not simulated) machines. The circuits data are given in Table 2.

| Qubits | Gates | CNOT | Time steps | Iterations | Repetitions | Success probability |
|--------|-------|------|------------|------------|-------------|---------------------|
| 4      | 519   | 291  | 396        | 3          | 8096        | 0.063               |
| 3 (odd)| 140   | 55   | 93         | 2          | 8096        | 0.4                 |
| 3 (even)| 59    | 24   | 39         | 2          | 8096        | NA                  |

Table 2: Circuit data for the undistributed computation with \(n = 4\) qubits and the two distributed circuits with \(n = 3\) qubits. The numbers in the table are for the transpiled circuits.

Fig. 5, shows the results for the undistributed Grover and Figs. 7 for the distributed Grover search. Clearly, the undistributed search failed, while the distributed search qubits succeeded in finding \(|1111\rangle\).

The results are in agreement with what one should expect for noisy machines that can handle limited depth. For more details about the expected and observed noise in the circuits, see Appendix B.

4.1 Undistributed Grover search

\[
|q_0\rangle \xrightarrow{H} |q_0\rangle \xrightarrow{Grover} |q_0\rangle \xrightarrow{Grover} |q_0\rangle
|q_1\rangle \xrightarrow{H} |q_1\rangle \xrightarrow{Grover} |q_1\rangle \xrightarrow{Grover} |q_1\rangle
|q_2\rangle \xrightarrow{H} |q_2\rangle \xrightarrow{Grover} |q_2\rangle \xrightarrow{Grover} |q_2\rangle
|q_3\rangle \xrightarrow{H} |q_3\rangle \xrightarrow{Grover} |q_3\rangle \xrightarrow{Grover} |q_3\rangle
\]

Figure 4: The undistributed Grover search circuit with \(n = 4\) qubits.

The optimal number of Grover iterations with \(n = 4\) is \(r = (\pi/4)\sqrt{16} = \pi \approx 3\) and the
(theoretical) success probability is \(0.96\). The circuit produced the histogram in Fig. 5 and manifestly failed to identify the target \(|1111\rangle\).

It is useful to extract from the histogram the frequencies of bit-flip error in the target \(|1111\rangle\). This is shown in Table 3. The data agree with the Binomial distribution up to \(\pm \sigma\). The large error probability for bit-flip, \(p_0 \approx 0.50\), is the reason for the failure of the search. \(p_0\) is in reasonable agreement with the estimates based on the machine data in Table 1, see Appendix B.

| # errors | 0  | 1  | 2  | 3  | 4  |
|-----------|----|----|----|----|----|
| Frequency | 0.06 | 0.26 | 0.38 | 0.25 | 0.06 |
| Binomial \([n = 4, p_0 \approx 0.50]\) | 0.06 | 0.25 | 0.38 | 0.25 | 0.06 |

Table 3: The bit flip error frequency in the search for the target \(|1111\rangle\) is well approximated by the Binomial distribution.

### 4.2 Distributed Grover search

The distributed Grover search runs with two Oracles, one for the ‘odd’ subspace and the other for the ‘even’ subspace, both with \(n = 3\) qubits. The target \(|1111\rangle\) is encoded in the odd Oracle and the even Oracle is “empty”. The optimal number of iterations in the odd circuit is \(r = (\pi/4)\sqrt{8} \approx 2\) with success probability \(p \approx 0.95\). The circuits are shown in Fig. 6.

The left histogram in Fig. 7 clearly identifies the correct answer \(|111\rangle\) (corresponding to \(|1111\rangle\)). The even circuit has no marked element\(^4\) and this is borne out by the histogram on the right.
Figure 6: The distributed circuits with $n = 3$ for the even/odd arguments.

Figure 7: The two histograms for the distributed search with $n = 3$ qubits. The sampling error is $\pm 0.011$. The left histogram correctly identified $|111\rangle$. The right histogram did not single out any state, as it should.

It is instructive to look at the probability of bit flip error. The histogram in Fig. 7 leads to Table 4. The agreement with a Binomial distribution is only qualitative. The small bit flip error $p_0 \approx 0.30$ is why the search succeeded. $p_0$ can be estimated from the machine data in Table 1, see Appendix B.

| # errors | 0     | 1     | 2     | 3     |
|----------|-------|-------|-------|-------|
| Frequency|       |       |       |       |
| Binomial [$n = 3, p_0 \approx 0.30$] | 0.35  | 0.44  | 0.19  | 0.03  |

Table 4: The bit flip error frequencies in the distributed search.

5 Distributed Simon’s algorithm

5.1 Period finding and the $Z_2^n$ Fourier transform

$p = \sin^2((2r + 1)\theta) \approx 0.96$ where $p = \sin \theta = 1/\sqrt{N}$

Although the formal optimum for $M = 0$ is $r = \infty$, the optimal number of Grover iterations is actually $r = 0$.  

8
Consider $f : \{0,1\}^n \mapsto \{0,1\}$, a Boolean function periodic with period $s \neq 0$ under bit-wise addition
\[
f(x \oplus s) = f(x) \quad \forall x \in \{0,1\}^n,
\]
(5.1)

Task: Find $s$. This is a special case of Simon's problem [14].

Period finding is the business of Fourier transforms. For the case at hand, this is the $Z_2^n$ Fourier transform. The circuit in Fig. 8 reduces period finding to solving a set of linear equations. Chasing the state $|0\rangle \otimes n$ through the circuit in Fig. 8, using the identity
\[
H \otimes n |0\rangle = \frac{1}{\sqrt{N}} \sum_y (-1)^{x \cdot y} |y\rangle
\]
(5.2)

one finds
\[
H \otimes n |0\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \xrightarrow{(-)^f} \frac{1}{\sqrt{N}} \sum_x (-1)^{f(x)} |x\rangle
\]
\[
= \frac{1}{2\sqrt{N}} \sum_x (-1)^{f(x)} (|x\rangle + |x \oplus s\rangle)
\]
\[
H \otimes n \xrightarrow{H} \frac{1}{2N} \sum_y \left( \sum_x (-1)^{f(x)+x \cdot y} \right) (1 + (-1)^{x \cdot y}) |y\rangle
\]
(5.3)

$g(y)$ is the $Z_2^n$ Fourier transform of $(-1)^f$, As such it is localized on arguments related to the period as can be seen from
\[
2g(y) = \sum_x (-1)^{f(x)+x \cdot y} + \sum_x (-1)^{f(x \oplus s)+(x \oplus s) \cdot y} = g(y) (1 + (-1)^{x \cdot y})
\]
(5.4)

It follows that:
\[
g(y) = w_y \delta (s \cdot y)
\]
(5.5)

$w_y \in \mathbb{Z}$ is the “weight” of the delta function. Inserting this to Eq. (5.3) gives for the state exiting the circuit
\[
\frac{1}{N} \sum_{y \mid y} w_{y_j} |y_j\rangle
\]
(5.6)

The outgoing state is therefore a linear combination of solutions of $s \cdot y = 0 \mod 2$. The period is determined as the solutions of the linear system
\[
y_j \cdot s = 0 \mod 2, \quad w_{y_j} \neq 0
\]
(5.7)

The random outcomes of the measurement of the quantum circuits give $y_j$. In the case there are $n$ independent vectors $y_j$, Eq. (5.7) has a unique solution $s = 0$. In the case that $n - 1$ vectors are independent there is a unique non-trivial period $s$, etc. In the case that the sum in Eq. (5.6) has a single term $w_0$ for $y = 0$, Eq. (5.7) trivializes and any $s$ is a solution: $f$ is a constant.
Remark 5.1. The weights \( \{w_y\} \) are (at most) \( N \) integers that satisfy Pythagoras

\[
\sum_{y \neq 0} w_y^2 = N^2 \tag{5.8}
\]

This imposes a (Diophantine) constraint on the allowed \( \{w_y\} \) which is independent of the function \( f \).

Example 5.2. With \( N = 4 \), the solutions \( \{w_y\} \) of Eq. (5.8) are

\[
\{ |4|, \ |2|, |2|, |2| \} \tag{5.9}
\]

Not all of the solutions are realized as Fourier transforms of (the phases of) Boolean functions. In fact, there are 16 Boolean functions of 2 bits:

- 2 constant functions \( f(x_1, x_2) = 0 \) and \( f(x_1, x_2) = 1 \) where \( s \) is arbitrary.
- 6 balanced functions where 1 has two pre-images. These have a single non-trivial period.
- The 4 functions where 1 has a single pre-image and 4 where 1 has three pre-images are a-periodic with \( s = 0 \).

The Fourier transforms of the phase functions \( (-1)^f \) are

- \( \pm 4\delta(y) \) for the constant functions.
- \( \pm 4\delta(y - j), \ j \in 1, 2, 3 \) for the balanced functions
- \( \pm 2(1 - 2\delta(y - j)), \ j \in 0, 1, 2, 3 \) for the a-periodic.

Consider the distributed algorithm in the case of a unique \( s \neq 0 \). Suppose first that \( s \) is even (e.g. \( s = 10 \)). The distributed Oracle reduces to the problem for \( n - 1 \) qubits for the even (odd) Oracles. This allows to determine \( s \) after \( O(n) \) queries. In the case that \( s \) is odd, \( x \) and \( x \oplus s \) have different parity. The \( n - 2 \) queries of the distributed algorithm will give the trivial result \( 2s = 0 \). One then needs to try again with a different notion of even-odd, per Eq. (2.3). If the new notion of even-odd gives \( s \) even the next \( n - 2 \) queries will determine \( s \) after a total of \( 3(n - 2) \) queries. If \( s \) is odd, we need to repeat the process. The complexity of an algorithm is determined by the worst case corresponding to \( n \) repetitions. This gives

\[
O(n^2) \tag{5.10}
\]

Remark 5.3. Similar arguments apply for the standard Simon algorithm for \( f : \{0, 1\}^n \mapsto \{0, 1\}^n \) (which is represented by a quantum circuit acting on \( 2n \)-qubits). Consider two quantum circuits each corresponding to the odd and even sub-spaces, defining functions \( f_c, f_o : \{0, 1\}^{n-1} \mapsto \{0, 1\}^n \) (each represented by a quantum circuit acting on \( 2n-1 \)-qubits). Following similar steps as in the paragraph above, we find that complexity in this case is also \( O(n^2) \).

In summary: The complexity of the period finding of a phase Oracle is:

- \( O(N) \) classically: The cost of the \( \mathbb{Z}_2^n \) Fourier transform.
- \( O(n) \) for the \( n \) qubits quantum circuit.
- \( O(n^2) \) for the distributed quantum circuit.

\(^5\)In the sense that for example for the first set \( w_y = |4| \) meaning either \( w_y = +4 \) or \( w_y = -4 \).
A Boolean function \( f \) is called balanced if
\[
\sum_x (-1)^{f(x)} = \sum_x (1 - 2f(x)) = 0 \tag{6.1}
\]
There are
\[
\binom{N}{N/2}, \quad N = 2^n \tag{6.2}
\]
balanced functions. A number which is super-exponentially large. There are, of course, only two constant Boolean functions: \( f \equiv 0 \) and \( f \equiv 1 \).

The Deutsch-Jozsa task is: Given the promise that \( f \) is either constant or balanced, determine which it is.

If no error is tolerated, one needs \( N/2 + 1 \) classical queries of \( f \). If one is satisfied with a correct answer with high probability, then few queries suffice. Indeed, the probability that \( k \) random queries of a balanced functions have the same image under \( f \) is
\[
p = \left(1 - \frac{N}{2N-1}\right) \cdots \left(1 - \frac{N}{2N-k+1}\right) \tag{6.3}
\]
When \( k \ll N \) one has
\[
p \approx \left(\frac{1}{2}\right)^k \tag{6.4}
\]
If one tolerates \( \epsilon \) error probability
\[
k = O(\log 1/\epsilon) \tag{6.5}
\]
queries suffice.

The Deutsch-Jozsa circuit of Fig. 8 outputs
\[
\text{Prob}(y = 0) = \left| \frac{1}{N} \sum_x (-1)^{f(x)} \right|^2 = \begin{cases} 1 & f \in \text{const} \\ 0 & f \in \text{balanced} \end{cases} \tag{6.6}
\]
and determines \( f \), with no error, with a single query (assuming that the quantum gates are error free and the \( f \) is indeed either balanced or constant).

Now consider the corresponding distributed Deutsch-Jozsa . The even and odd parts of a constant function are still constant functions. But, the even and odd parts of a balanced function need not be (two) balanced functions. The distribution of 1 in the even function is the same as randomly drawing \( N/2 \) stones from two urns with \( N/2 \) stones each, all 1 or all 0. The probability distribution for finding \( k \) 1’s in the even sequence is
\[
\text{Prob}(k) = \binom{N/2}{k} \binom{N/2}{N/2 - k} / \binom{N}{N/2} \tag{6.7}
\]
We are interested in the probability that the distributed circuit will (mis) identify a balanced function as constant. By Eq. 6.6 this is related to the expectation of
\[
\left| \frac{1}{N} \sum_x (-1)^{f_e(x)} \right|^2 = \left(1 - \frac{2}{N} \sum_x f_e(x)\right)^2 = 1 - \frac{4}{N} \sum_x f_e(x) + 4 \left(\frac{1}{N} \sum_x f_e(x)\right)^2 \tag{6.8}
\]
\[\text{The formula follows from repeated application of the fact that in an urn with } W \text{ white stones and } B \text{ black stone, the probability for picking a black stone is } B/(B+W).\]
Evidently
\[ \mathbb{E}\left(\sum_x f_e(x)\right) = \sum_k k \cdot \text{Prob}(k) = \frac{N}{4} \]  
(6.9)

As \( f_e(x) \) and \( f_e(y) \) are independent for \( x \neq y \) and \( f_e^2(x) = f_e(x) \) we also have
\[ \mathbb{E}\left(\sum_x f_e(x)\right)^2 = \mathbb{E}\left(\sum_{x,y} f_e(x)f_e(y)\right) = \mathbb{E}\left(\sum_x f_e(x)\right)^2 = \frac{N}{4} \]  
(6.10)

It follows that the expectation values that a single query of the distributed circuit will make the mistake of identifying a balanced \( f \) as constant is
\[ \mathbb{E}(y = 0|f = \text{balanced}) = \frac{1}{N} \]  
(6.11)

Comparing with Eq. (6.5) we see that one needs \( O(n) \) classical queries to get the same margin of error as a single quantum query.

In summary: The complexity of the Deutsch-Jozsa problem is
- 1 + \( N/2 \) classical queries for a deterministic result.
- A single quantum query (for an error-free circuit).
- A single query of a distributed quantum circuits for \( O(1/N) \) error.
- \( O(n) \) classical queries for \( O(1/N) \) error.

The first two entries imply an \( O(N) \) quantum advantage and the last two entries a \( O(n) \) advantage of distributed quantum computing.

7 Conclusion

Distributed quantum computing with classical communication on ideal devices is, in general, inferior to undistributed quantum computing. However, in the context of the currently available noisy small computers that offer no error corrections, distributed computing offers two advantages: Amalgamating qubits resources and shallow circuits with significant noise reduction.

Acknowledgment

We thank Eyal Bairey, Shay HaCohen-Gourgy, Oded Kenneth and Netanel Lindner for helpful discussion and E.B. for pointing out [9]. We thank the referees for pointing out [5, 12].

Appendix A The DNF of quantum circuits

For the sake of simplicity and concreteness we illustrate an algorithms that works for any Boolean function \( f : \{0,1\}^n \rightarrow \{0,1\}^m \) by considering the special case \( n = 2 \) and \( m = 1 \).
A.1 From the truth table to DNF [1]

The Disjunctive Normal Form [7] of a Boolean function can be calculated directly from the truth table of the function. Consider:

| \( x_0 \) | \( x_1 \) | \( f(x) \) |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

The DNF form of \( f \) is given by

\[
f_{\text{DNF}}(x) = \bar{x}_0 \cdot \bar{x}_1 + x_0 \cdot x_1
\]  

(A.1)

where \( \bar{x} \) denotes \( \text{NOT}(x) \).

We only need to consider the rows where \( f(x) = 1 \). For every such row we build out of the arguments \( x_0, x_1 \) a Boolean statement made of only NOT and AND operations (conjunctive) so that an argument that takes the value 1 is written as is, while argument with value 0 is negated. For the example:

- The first row is described by \( \bar{x}_0 \cdot \bar{x}_1 \).
- The third row is described by \( x_0 \cdot \bar{x}_1 \).

**Generalization:** In the case of general \( n \) do the same with \( n \) variables. For a different \( m \) repeat the process for each \( f_i(x) \).

The DNF extracted from the truth table is not, in general, the simplest DNF formula of the function. For example \( f \) of Eq. (A.1) can be written more simply as:

\[
f_{\text{DNF}} = \bar{x}_1
\]

The optimal DNF formula can be calculated using Karnaugh maps [11].

A.2 From DNF to quantum circuit [4]

The quantum circuit for any \( f : \{0,1\}^n \rightarrow \{0,1\}^m \) can be built with \( X \) gates and \( C^\otimes n X \) gates. To see this recall that Toffoli gate [15] gives the conjunction of its arguments \( f_T(x_0, x_1) = x_0 \cdot x_1 \).

The quantum circuit for calculating the function in Eq. (A.1) is shown in Fig. 9. (It can be simplified using \( XX \equiv 1 \).)
Generalization: For the case of \( n \) conjuctions, a similar constructions works with \( C^\otimes n X \) replacing Toffoli:

\[
\begin{array}{c}
|x_0\rangle \\
\vdots \\
|x_n\rangle \\
|c\rangle
\end{array}
\begin{array}{c}
|\rangle \\
|c + x_0 \cdot x_1 \cdots x_n\rangle
\end{array}
\]

For Boolean functions whose output is a string of \( m \) bit one simply adds additional target qubits.

A.3 Phase Oracles

The same method can be used to construct the phase Oracle for \((-1)^{f(x)}\). This is done by replacing \( C^\otimes n X \) by \( C^\otimes (n-1) Z \) gates, for example the circuit for the phase gate of the function in Eq. (A.1) is shown in Fig. 10.

\[
|0\rangle \iff X \iff |0\rangle
|1\rangle \iff X \iff |\bar{1}\rangle
|0\rangle \iff X \iff |0\rangle \iff X \iff |\bar{1}\rangle
\]

Figure 10: The phase gate for the function in Eq. (A.1).

Appendix B The experiment: Additional details

B.1 Splitting the Grover Oracle

Alice and Bob each get the same \( n \)-qubit Oracle. Alice uses the Oracle as is. Bob, uses the following algorithm to split the Oracle to the two computers \( B_1 \) and \( B_2 \):

\begin{algorithm}
\hspace{1cm} Result: The binary index of a desired element from the whole set.
1 BobGrover \((U_f)\)
2 \( U_{odd}, U_{even} \leftarrow \text{Divide}(U_f); \)
3 \( res_1 \leftarrow \text{Grover}(B_1, U_{even}); \)
4 \( res_2 \leftarrow \text{Grover}(B_2, U_{odd}); \)
5 if \( U_{even}(res_1) == 1 \) then
6 \hspace{1cm} return \( res_1 + '0'; \)
7 else \hspace{1cm} return \( res_2 + '1'; \)
\end{algorithm}

B.2 A depolarizing channel model for the errors

A simple model of the noise in circuits is in terms of depolarizing channels\(^7\):

\[
\rho \mapsto \lambda \rho + \frac{1-\lambda}{2} \mathbb{I}, \quad -1/3 \leq \lambda \leq 1
\]

\(^7\)The condition on \( \lambda \) guarantees complete positivity, see, Wikipedia
The probabilities that a qubit $|\psi\rangle$ gives the correct (1) /incorrect (0) answers are

$$p_0 = \frac{1 - \lambda}{2}, \quad p_1 = \frac{1 + \lambda}{2}, \quad (B.2)$$

Assuming independence, the probability for $m$ bit-flip errors in $n$ qubits register is

$$\binom{n}{m} p_0^m p_1^{n-m} \quad (B.3)$$

The expected number of errors in the register is

$$np_0 \quad (B.4)$$

and the probability for the correct answer in all $n$ qubits is

$$\left(\frac{1 + \lambda}{2}\right)^n \quad (B.5)$$

### B.3 Decoherence and gate errors

The decoherence associated with each time step is represented by a polarizing channel with $\lambda \mapsto \mu_c$ and the gates errors gate by a polarizing channel with $\lambda \mapsto \mu_g$. The composition of polarizing channels is ordinary multiplication. It follows that for a circuit with $T$ time steps and $N_g$ noisy gates, is a channel where

$$\lambda \approx \mu_c^T \mu_g^{N_g} \quad (B.6)$$

Since the dominant error is in the 2-qubits gates, $N_g$ is the number of 2-qubits gates.

From Table 1

$$\log \mu_c = -\frac{R}{T_c} \approx -3 \times 10^{-3} \implies \mu_c \approx 0.997, \quad \mu_g = 1 - \epsilon \approx 0.994 \quad (B.7)$$

An estimate for $\lambda$ is then

$$\lambda \approx (0.997)^T \cdot (0.994)^{N_g} \quad (B.8)$$

This allows to compare the experimentally observed probability for bit flip error $p_0$ computed via Eq. (B.4) and the data in Tables 3, 4 with the expected bit flip error $\bar{p}_0$ computed via Eqs. (B.2) and (B.8) using the IBM machine data in Table 1. The qualitative agreement between $p_0$ and $\bar{p}_0$ gives some support to the noise model as depolarizing channel.

| Qubits | $p_0$ | $\bar{p}_0$ | $\lambda$ |
|--------|-------|-------------|----------|
| 4      | 0.50  | 0.47        | 0.05     |
| 3 (odd)| 0.30  | 0.23        | 0.54     |

Table 5: The bit flip error $p_0$ computed from the histogram and $\bar{p}_0$ computed from the Machine data.

*Since $T, N_g = O(1000)$ we need to keep 3 significant figures.*
References

[1] David Hilbert; Wilhelm Ackermann. *Principles of Mathematical Logic*. American Mathematical Soc, 1999.

[2] J.-H. Bae, Paul M. Alsing, Doyeol Ahn, and Warner A. Miller. Quantum circuit optimization using quantum karnaugh map. *Scientific Reports*, 10(1):15651, Sep 2020.

[3] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.

[4] Yu. I. Bogdanov, N.A. Bogdanova, D.V. Fastovets, and V.F. Lukichev. Representation of boolean functions in terms of quantum computation. *Proc. SPIE 11022, International Conference on Micro- and Nano-Electronics 2018, 110222R*, arXiv:1906.06374, 2019.

[5] Sergey Bravyi, Graeme Smith, and John A Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6(2):021043, 2016.

[6] JI Cirac, AK Ekert, SF Huelga, and Chiara Macchiavello. Distributed quantum computation over noisy channels. *Physical Review A*, 59(6):4249, 1999.

[7] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*, page 153. Cambridge University Press., 1990.

[8] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

[9] Vedran Dunjko, Yimin Ge, and J. Ignacio Cirac. Computational speedups using small quantum devices. *Phys. Rev. Lett.*, 121:250501, Dec 2018.

[10] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC)*, May 1996, pages 212-219. arXiv:quant-ph/9605043, 1996.

[11] M. Karnaugh. The map method for synthesis of combinational logic circuits. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 72(5):593–599, 1953.

[12] Tianyi Peng, Aram W Harrow, Maris Ozols, and Xiaodi Wu. Simulating large quantum circuits on a small quantum computer. *Physical Review Letters*, 125(15):150504, 2020.

[13] Peter W Shor. Introduction to quantum algorithms. In *Proceedings of Symposia in Applied Mathematics*, volume 58, pages 143–160, 2002.

[14] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997.

[15] Tommaso Toffoli. Reversible computing. In Jaco de Bakker and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 632–644, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.