Supervisor: Prof. Mike Payne
Review Title: "Interpretations of Quantum Mechanics and Topics in Quantum Information"

Algorithms for Quantum Computing, An Overview

Abstract

Quantum computers are machines capable of outperforming their classical counterparts in certain tasks. This is achieved through the design of efficient quantum algorithms which must adhere to a new paradigm of computing. A formulation of the novel quantum approach, characterised by quantum parallelism, interference, and non-deterministic results, is presented, and its implementation in the Deutsch-Jozsa, Grover's and Shor's algorithms is discussed.

1 Introduction

The field of quantum computing concerns itself with machines capable of performing tasks using methods which are physically unavailable to any classical computer. These additional capabilities allow for a range of problems to be solved in a more efficient manner, for instance in the areas of cryptography, search and optimisation, and quantum system simulation.[1] These applications have transpired through the development of quantum algorithms, and the aim of this article is to provide an overview of the best-known of these – the Deutsch-Jozsa, Shor's, Grover's algorithms.

An important difference exists between the approaches taken during the design of algorithms for classical and quantum computers. David Deutsch suggested[2,3] that in order to design efficient quantum algorithms, a change in the paradigm of computing is required. Only then would an advantage over the classical approach be gained. Therefore, this review also aims to provide a description of how the three algorithms above implement the change in approach, exemplifying its importance for the design of future methods.

An explanation of the necessary changes in the computing paradigm is presented in the following section. In section 3 the three algorithms are described, and the conclusions are formulated in the final section.

2 The Quantum Paradigm of Computing

One of the unique properties of quantum physics is that systems can exist in superpositions of observable states. David Deutsch realised[2] that this characteristic can be used by quantum computers to perform calculations on multiple inputs simultaneously, in hand leading to an increase in performance. Consider for instance a device which evaluates a given function f acting on some input space $I_f \equiv \{x_i\}$. In order to get the value of the function for each possible input using a classical implementation, the calculation would have to be performed as many times as there are elements in I_f. Effectively, only one evaluation of $f(x)$ can be performed per operation of the device. Consider now a quantum implementation. Due to its quantum nature, inputs consisting of superpositions of the possible elements in I_f can be fed into the device. Therefore, upon operating it

on such an input state, an output consisting of multiple evaluations of the function f can be generated, all while running the machine only once. This property of quantum computers is called quantum parallelism, and is a fundamental part of the operation of any quantum algorithm. It is represented pictorially in Figure 1.

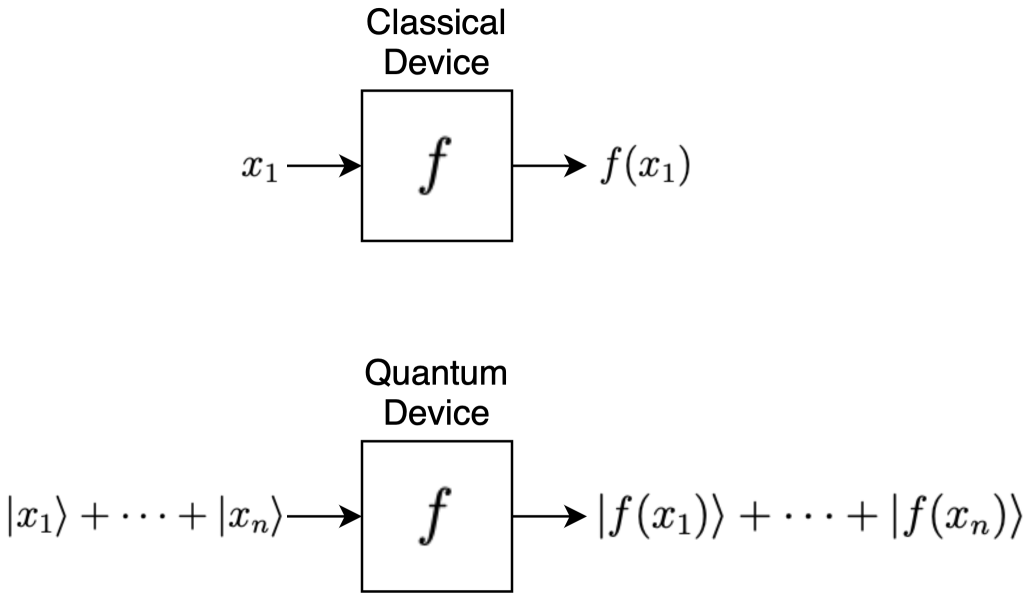

Figure 1: Diagrams depicting the operation of the function f implemented on a classical (above) and a quantum (below) device. On the classical device one is limited to a single evaluation of the function per run. However, on the quantum device an input state can correspond to a superposition of different elements from the function's input space. This is made clear using the bra-ket notation. Hence, multiple evaluations of f can be performed simultaneously during one run of the device.

There is however a subtlety in the way quantum parallelism must be utilised due to the measurement problem in quantum mechanics. In particular, consider the output state from the quantum device in Figure 1. If a measurement is performed on this superposition of function evaluations $f(x_1), \ldots, f(x_n)$, the state will collapse onto a single output, say $f(x_i)$. In such cases, no advantage over simply performing a classical evaluation of the function for the input x_i will be gained. To overcome this, it is suggested[2] that quantum algorithms will be most effective when they seek global structure in the respective system, which in the specific case above would be the function f. This can be achieved through another property of the theory – quantum mechanical interference. In general, algorithms will manipulate the superpositions during their operation, causing the constituent states to interfere appropriately, thereby encoding the desired result in a quantity which can later be measured.

Finally, when performing a measurement on a quantum system, there is often an intrinsic uncertainty in the outcome. Therefore, it is possible that quantum algorithms may not give deterministic answers to the questions they are designed to answer.[3] In particular, it is reasonable to expect that the outcomes of separate executions of a given algorithm may differ. This implies that several runs may be required in order to solve a problem with certainty. Nonetheless, efficient quantum devices aim to decrease the number of runs as much as possible, i.e., make the scenario of obtaining the desired outcome

as likely as possible.

Overall, the quantum paradigm of computing can be summarised as follows. A quantum algorithm:

- Will make use of quantum parallelism in order to perform operations on multiple inputs simultaneously;
- Will likely look for a global structure in the system, and use quantum mechanical interference to encode it in a measurable property of the final state, thereby overcoming the measurement problem;
- May not give deterministic answers, in that multiple executions may be required to obtain the desired result with certainty.

In the next section a description is given of how specific quantum algorithms incorporate the points above to gain advantage over their classical counterparts.

3 Quantum Algorithms

3.1 The Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm was proposed by David Deutsch and Richard Jozsa in 1992,[3] and later generalised in 1998,[4] which solves the following problem. Consider the function f with input space $I_f \equiv \{x_i\}$ containing $N = 2^n$ elements, and output space $O_f \equiv \{0, 1\}$. It is known that f is either constant, meaning that the outputs $f(x_i)$ are the same for all x_i, or balanced, which means that exactly half of the outputs are 0, and the other half are 1. The aim is to determine the type of the function. Although it finds limited practical use, the algorithm is one of the first to achieve quantum supremacy over its classical version, and is therefore a useful proof of concept.

Before describing the algorithm, it is important to explain how functions f are implemented on quantum computers. The operation of f on a given state of the system is performed using an operator, $\hat{U}_f$, the inner workings of which are not accessible. This is a crucial point, for if the details of $\hat{U}_f$ are known, the properties of the function can be inferred without running the algorithm. The function f is said to be implemented as a quantum oracle, or black box. One general definition[5] of such an oracle is:

$$\hat{U}_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle , \tag{1}$$

where $\left|x\right\rangle \left|y\right\rangle$ is short-hand notation for the tensor product $\left|x\right\rangle \otimes \left|y\right\rangle$. The operation $\oplus$ denotes digit-wise addition modulo 2. For instance, $6 \oplus 5 = 3$, since the base-2 representations of $6, 5$, and 3 are $110, 101,$ and 011, respectively. An important property of the definition in (1) is that the new state $\left|x\right\rangle \left|y \oplus f(x)\right\rangle$ retains information about both the input x and output $f(x)$.

A description of the algorithm is now given. An initial state is created out of $n + 1$ qubits. A qubit is the quantum mechanical equivalent of the bit used in classical computing. It is a two-level system, spanned by the basis $\{\left|0\right\rangle , \left|1\right\rangle\}$. A state consisting of

multiple qubits is just the tensor product of the states of each individual qubit. For instance, the overall state of the two qubits, each in state $|0\rangle$, is $|0\rangle \otimes |0\rangle \equiv |0\rangle\,|0\rangle \equiv |0\rangle^{\otimes 2}$. In the case of the Deutsch-Jozsa algorithm, the initial state of the $n+1$ qubit system is:

$$|\psi_0\rangle = |0\rangle^{\otimes n}\,|1\rangle\,, \tag{2}$$

which means that the first n qubits are initialised in states $|0\rangle$, whereas the last qubit is in state $|1\rangle$. The first n qubits are said to form the first quantum register of our system, and the $(n+1)^{\text{st}}$ qubit – the second quantum register.

The state in (2) is then mapped onto a new state, $|\psi_1\rangle$, through the application of Hadamard transforms on each qubit. The Hadamard transform, labelled with the operator $\hat{H}$, is defined for two-level systems as follows:

$$\hat{H}\,|0\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle)\,,$$
$$\hat{H}\,|1\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle - |1\rangle)\,.$$

It can be viewed as a change of basis, in particular a rotation, for the two-level system, as shown in Figure 2.

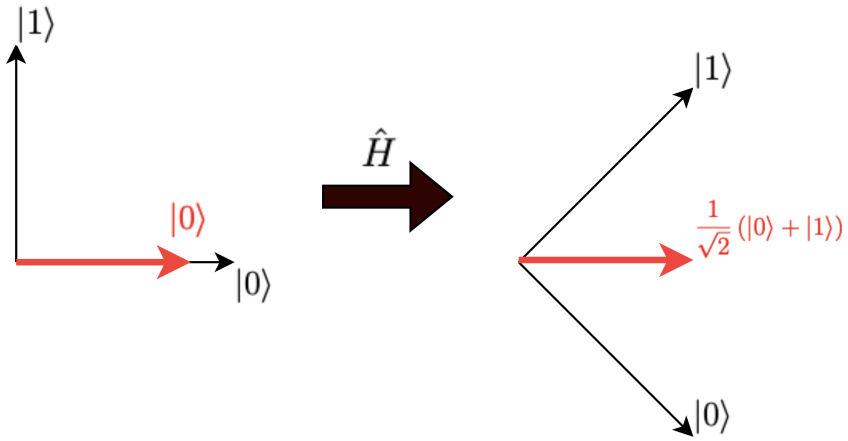

Figure 2: Pictorial representation of the action of the Hadamard transform $\hat{H}$ on the qubit state $|0\rangle$ (shown in red). The transform rotates the basis spanning the two-level qubit system, drawn with black arrows, 45° clockwise, taking the state $|0\rangle$ to $\frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle)$. Note to improve clarity that the arrows depicting the basis vectors and the state vectors are not drawn to scale.

Using the definition of $\hat{H}$ above, the new state is:

$$|\psi_1\rangle = \hat{H}^{\otimes(n+1)}\,|\psi_0\rangle = \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\,(|0\rangle - |1\rangle), \tag{3}$$

where $\hat{H}^{\otimes(n+1)}$ denotes application of $\hat{H}$ on each qubit in $|\psi_0\rangle$. In (3) short-hand notation has been used to represent the tensor product states $|x\rangle$ appearing in the sum. For instance, the state $|5\rangle$ is equivalent to $|101\rangle \equiv |1\rangle\,|0\rangle\,|1\rangle$, since the binary representation of 5 is 101. The remaining $n-3$ qubits (which are in states $|0\rangle$) are neglected for clarity. It is important to note that the first register in the $|\psi_1\rangle$ in (3) corresponds to a superposition

of all 2^n possible inputs to the function f, where each input is labelled with a unique integer from 0 to $2^n - 1$.

The next step is to act on $|\psi_1\rangle$ with the quantum oracle $\hat{U}_f$:

$$\begin{aligned}
|\psi_2\rangle = \hat{U}_f |\psi_1\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} \hat{U}_f |x\rangle (|0\rangle - |1\rangle) \\
&= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\
&= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle),
\end{aligned}$$

where the last line follows because $f(x)$ is either 0 or 1. An important point is that the function has been evaluated for each of its 2^n possible inputs simultaneously. This exemplifies how quantum parallelism is employed in the Deutsch-Jozsa algorithm. At this point the $(n+1)^{\text{st}}$ qubit can be ignored, i.e., just the state $|\psi_2^{(1)}\rangle$ of the first register is considered:

$$|\psi_2^{(1)}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle . \tag{4}$$

The final step in the algorithm is to again apply a Hadamard transform to each of the n qubits in $|\psi_2^{(1)}\rangle$. In fact, since $\hat{H}^2 = \hat{I}$, this is equivalent to returning to the original bases of our two-level systems. The resulting state is:

$$\begin{aligned}
|\psi_3^{(1)}\rangle = \hat{H}^{\otimes n} |\psi_2^{(1)}\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \hat{H}^{\otimes n} |x\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left[\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \right] \\
&= \sum_{y=0}^{2^n-1} \left[\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y} \right] |y\rangle \\
&\equiv \sum_{y=0}^{2^n-1} A(y) |y\rangle ,
\end{aligned}$$

where the expansion coefficient $A(y) \equiv \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y}$ has been defined, with $x \cdot y = x_0 y_0 \oplus \cdots \oplus x_{n-1} y_{n-1}$ being the modulo 2 sum of the digit-wise product of x and y in their binary representations. The result

$$\hat{H}^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \tag{5}$$

is an equivalent definition for the n-qubit Hadamard transform.[4]

A measurement of the final state $|\psi_3^{(1)}\rangle$ in the first register of the system is now performed. The probability of measuring the state $|\psi_0^{(1)}\rangle = |0\rangle^{\otimes n}$ is:

$$P_0 = |A(0)|^2 = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2 = \begin{cases} 0 & \text{if } f \text{ is balanced,} \\ 1 & \text{if } f \text{ is constant.} \end{cases} \tag{6}$$

In other words, if the system collapses onto the state $|\psi_0^{(1)}\rangle = |0\rangle^{\otimes n}$, f is constant. Otherwise, f is balanced.

Overall, the Deutsch-Jozsa quantum algorithm solves the problem at hand with just one query to the quantum oracle. On a classical computer the solution requires $2^{n-1}+1$ queries to the function f in the worst case. In other words, the complexity of the algorithm scales linearly with the input size of the question N. The quantum approach solves the problem much faster, namely with an exponential increase in performance.

Moreover, it is apparent how the Deutsch-Jozsa algorithm follows the quantum paradigm of computing. It utilises quantum parallelism by creating a superposition state of all possible inputs to the function, $|\psi_1\rangle$, and evaluates the function f for all of them simultaneously to create the state $|\psi_2\rangle$. Quantum mechanical interference is then used through Hadamard transforms, encoding the desired global property of the function, namely whether f is constant or balanced, in a single expansion coefficient, $A(0)$, which can later be inferred through a single measurement on the final state. In particular, the constituent states of the superposition $|\psi_2^{(1)}\rangle$ interfere constructively to set $A(0) = 1$ if f is constant, or destructively to set $A(0) = 0$ if f is balanced. With respect to the third point in the quantum paradigm, the algorithm in fact solves the problem with certainty, meaning that only one execution is required to reach the desired result.

3.2 Grover's Algorithm

Grover's algorithm was developed by Lov Grover in 1996[6] to solve the following problem. Consider an unstructured database of N elements. The goal is to find a specific entry, ω. Implementations of this algorithm on real quantum computers would find much practical use, for such unstructured database searches are common tasks. Moreover, variations of it may be used to find smallest values in lists, approximate mean values, determine graph connectivity, and match patterns, as performed by search functions in text-processing programs.[1]

Before running the algorithm, all the entries in the database are labelled with numbers ranging from 0 to $2^n - 1$, where $n = \lceil \log_2(N) \rceil$. If N is not a power of 2, additional elements are included in the database until the required size is reached. This is important since each entry is later represented as a separate state in a system of qubits, which for n qubits amounts to 2^n states.

An initial state is then created out of n qubits, all being in state $|0\rangle$:

$$|\psi_0\rangle = |0\rangle^{\otimes n}. \tag{7}$$

Similarly to the Deutsch-Jozsa algorithm, $|\psi_0\rangle$ is mapped on to $|\psi_1\rangle$ through the application of Hadamard transforms on each of the qubits:

$$|\psi_1\rangle = \hat{H}^{\otimes n} |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \,. \tag{8}$$

In (8) the short-hand notation from (3) for the tensor product states $|x\rangle$ is again used. Due to the initial labelling, $|\psi_1\rangle$ represents a superposition state of all the possible entries in the database.

A set of operators which will act on the system states while the algorithm is being executed are now introduced. The first of them is $\hat{U}_\omega$, which acts as the quantum oracle introduced in section 3.1. It gives information about whether the state it is applied to corresponds to the entry ω or not. Of course, the exact operation of the oracle is not known, for if it was, ω could be found without performing any queries, for instance by inspecting $\hat{U}_\omega$'s matrix representation. An example of such a quantum black-box is:

$$\hat{U}_\omega = \hat{I} - 2\,|\omega\rangle\,\langle\omega| \,. \tag{9}$$

The n-qubit states $|x\rangle$ for $x \in [0, 2^n - 1]$, corresponding to all the entries in the database, form a complete set of eigenstates for $\hat{U}_\omega$. The state $|\omega\rangle$ has -1 as its eigenvalue, whereas all others have eigenvalues $+1$. Hence, $\hat{U}_\omega$ is used as a quantum oracle by inspecting its output when acting on a given state. In particular, if this output is -1, the system has collapsed to the desired state $|\omega\rangle$. The other operator required is $\hat{U}_{\psi_1}$:

$$\hat{U}_{\psi_1} = 2\,|\psi_1\rangle\,\langle\psi_1| - \hat{I}, \tag{10}$$

using $|\psi_1\rangle$ from (8).

The algorithm now proceeds by performing multiple Grover iterations on the state of the n-qubit system, starting from $|\psi_1\rangle$. One Grover iteration, $\hat{G}$, is defined as consecutive application of $\hat{U}_\omega$ and $\hat{U}_{\psi_1}$:

$$\hat{G} = \hat{U}_{\psi_1}\hat{U}_\omega. \tag{11}$$

The action of $\hat{G}$ on an arbitrary state of the system is interpreted geometrically.[7] From (8) the inner product

$$\langle\omega|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \neq 0. \tag{12}$$

Hence, $|\psi_1\rangle$ and $|\omega\rangle$ form a two-dimensional subspace, $\mathcal{V}$, of the whole n-qubit vector space. From the definitions of $\hat{U}_\omega$ and $\hat{U}_{\psi_1}$, $\hat{G}\,|\psi\rangle$ only differs from $|\psi\rangle$ in its components $\langle\omega|\psi\rangle$ and $\langle\psi_1|\psi\rangle$ along $|\omega\rangle$ and $|\psi_1\rangle$, respectively. In other words, if $|\psi\rangle \in \mathcal{V}$, then $\hat{G}\,|\psi\rangle \in \mathcal{V}$ as well. The geometrical interpretation of $\hat{U}_\omega$ is reflection through the vector $|\omega_\perp\rangle \in \mathcal{V}$,

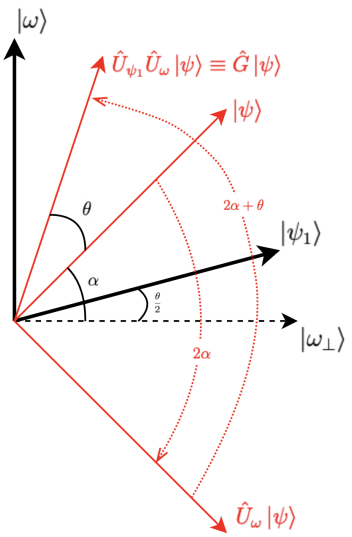

Figure 3: Geometrical interpretation of the Grover iteration. $|\psi\rangle$ is an arbitrary vector in the 2-dimensional subspace $\mathcal{V}$ spanned by $|\omega\rangle$ and $|\psi_1\rangle$. The operation of $\hat{U}_\omega$ rotates $|\psi\rangle$ by an angle 2α clockwise, whereas $\hat{U}_{\psi_1}$ rotates the resulting state $2\alpha+\theta$ counter-clockwise. Effectively, one Grover iteration $\hat{G} \equiv \hat{U}_{\psi_1}\hat{U}_\omega$ rotates the vector it acts on by an angle θ counter-clockwise.

orthogonal to $|\omega\rangle$. This is depicted in Figure 3 for the arbitrary state $|\psi\rangle \in \mathcal{V}$. Effectively, $|\psi\rangle$ is rotated by an angle 2α clockwise, where $\sin(\alpha) \equiv \langle\omega|\psi\rangle$. On the other hand, $\hat{U}_{\psi_1}$ is interpreted as a reflection through the vector $|\psi_1\rangle$, i.e., rotation of $\hat{U}_\omega |\psi\rangle$ by an angle $2\alpha + \theta$ counter-clockwise, where $\sin\left(\frac{\theta}{2}\right) \equiv \langle\omega|\psi_1\rangle = \frac{1}{\sqrt{2^n}}$. Overall, one Grover iteration amounts to a rotation by θ counter-clockwise, as shown in Figure 3.

The algorithm performs consecutive Grover iterations, starting from $|\psi_1\rangle$ until the resulting state of the system has greatest overlap with $|\omega\rangle$, i.e., $\langle\omega|\hat{G}^r|\psi_1\rangle$ is as large as possible, where r is the number of iterations. At that point, the probability of measuring an output of -1 when acting on the current state of the n-qubit system with the quantum oracle $\hat{U}_\omega$, and thereby finding the desired entry from the wavefunction collapse, will be maximised. From the geometrical interpretation, the angle between $\hat{G}^r |\psi_1\rangle$ and $|\omega\rangle$ is:

$$\beta(r) = \frac{\pi}{2} - r\theta - \frac{\theta}{2}. \tag{13}$$

Hence, the probability of finding ω through application of the quantum oracle after r Grover iterations is:

$$P_\omega(r) = |\langle\omega|\hat{G}^r|\psi_1\rangle|^2 = \sin^2\left(\left[r + \frac{1}{2}\right]\theta\right). \tag{14}$$

This probability is maximised, in the limit of large databases, i.e., large 2^n, after:

$$r_{\max} \approx \frac{\pi\sqrt{2^n}}{4} \tag{15}$$

Grover iterations.[8]

Overall, Grover's algorithm finds the correct entry with large probability after $\mathcal{O}(\sqrt{2^n})$ queries to the quantum oracle. On a classical computer, the most effective solution will require $2^n - 1$ queries in the worst case, resulting in a complexity of $\mathcal{O}(2^n)$. Therefore, the quantum approach provides a quadratic increase in performance over its classical counterpart.

Moreover, it is again apparent how the algorithm obeys the quantum computing approach. It uses quantum parallelism by creating a superposition state, $|\psi_1\rangle$, of all possible entries in the database, and performs operations on them simultaneously. In particular, these include Grover iterations which use quantum mechanical interference to increase the magnitude of the expansion coefficient $\langle\omega|\hat{G}^{r_{\max}}|\psi_1\rangle$ in the final state, corresponding to the desired entry ω. This technique is called amplitude amplification. However, the algorithm does not necessarily give a deterministic answer to the search. There is some probability of collapsing the n-qubit system on to a state different from $|\omega\rangle$ when performing the final measurement with the quantum oracle. In such cases, the algorithm must be executed again from the start. Nonetheless, the probability of measuring the desired outcome can be maximised significantly, as shown above.

3.3 Shor's Algorithm

Shor's algorithm was discovered by Peter Shor in 1994.[9] It consists of both a classical and a quantum component, which altogether perform prime factorisation. Just the

quantum part, which answers the following problem, is reviewed here. Consider a function $f(x) = a^x \pmod N$, where $a, N > 0$ and $x \geq 0$ are integers, with a and N being co-prime. This function is periodic with period r equal to the multiplicative order of a modulo N. The goal is to find r. This is an extremely important algorithm, for it can be used to break public-key cryptography schemes,[1] such as the RSA, which relies on the practical difficulty of finding the prime factors of large numbers on classical computers.

The quantum system used by the algorithm consists of $m + 1$ qubits, where $2^m \in [N^2, 2N^2)$. The first m qubits form the first register, and the remaining $(m+1)^{\text{st}}$ qubit – the second register. The system is prepared in the initial state:

$$|\psi_0\rangle = |0\rangle^{\otimes m} |1\rangle . \tag{16}$$

Similarly to the previous two algorithms, $|\psi_0\rangle$ is mapped on to $|\psi_1\rangle$ using Hadamard transforms, applied just to the qubits in the first register:

$$|\psi_1\rangle = (\hat{H}^{\otimes m} \otimes \hat{I}) |\psi_0\rangle = \sum_{x=0}^{2^m-1} |x\rangle |0\rangle , \tag{17}$$

where the short-hand notation from (3) to represent the tensor product states $|x\rangle$ is used. Then, $|\psi_1\rangle$ is acted on with the quantum implementation of the function f, i.e., the quantum oracle, which is defined as in (1):

$$\begin{aligned} |\psi_2\rangle = \hat{U}_f |\psi_1\rangle &= \sum_{x=0}^{2^m-1} \hat{U}_f |x\rangle |0\rangle \\ &= \sum_{x=0}^{2^m-1} |x\rangle |0 \oplus f(x)\rangle \\ &= \sum_{x=0}^{2^m-1} |x\rangle |f(x)\rangle . \end{aligned}$$

The state of the second quantum register is measured, in order to collapse the first register's state appropriately. In particular, if the $(m+1)^{\text{st}}$ qubit is measured to be in state $|f(x) = a\rangle$, the state of the first register will collapse on to:

$$|\psi_3^{(1)}\rangle = \sum_{x \in \{x_a\}} |x\rangle , \tag{18}$$

where $\{x_a\}$ is the set of all integers in $[0, 2^m)$ for which $f(x_a) = a$. This state has the useful global property that the numbers in $\{x_a\}$ are separated by the desired period r.

The last step is to apply the Quantum Fourier Transform (QFT)[9] to $|\psi_3^{(1)}\rangle$ in order to encode the period r into a measurable quantity of the final state $|\psi_4^{(1)}\rangle$:

$$|\psi_4^{(1)}\rangle = \mathcal{F}_Q \left[|\psi_3^{(1)}\rangle \right] , \tag{19}$$

where $\mathcal{F}_Q[\dots]$ denotes taking the QFT. A measurement of the state $|\psi_4^{(1)}\rangle$ in (19) will yield the correct period r with higher probability the more numbers are contained in the

set $\{x_a\}$. This is the reason for choosing m such that $2^m \in [N^2, 2N^2)$.

Overall, Shor's algorithm solves the period-finding problem with large probability after just one query to the quantum oracle. On a classical computer an equivalent algorithm would have to perform evaluations of the function for each of the 2^m integers forming the superposition in (17). Effectively, after taking into account the complexity of performing the QFT,[9] Shor's algorithm offers a nearly exponential increase in performance over its classical counterpart.

Additionally, it is again observed how the algorithm adheres to the quantum paradigm. Similarly to the previous two algorithms, it implements quantum parallelism when forming the superposition state $|\psi_1\rangle$, and applying the quantum oracle to all of its constituent states simultaneously. It employs quantum mechanical interference in its application of the QFT in order to encode the desired global property of $|\psi_3^{(1)}\rangle$, i.e., the period r, into a quantity explicitly measurable in $|\psi_4^{(1)}\rangle$. Finally, as with Grover's, Shor's algorithm does not necessarily give a deterministic answer for r. Nonetheless, the probability of measuring the correct value is maximised by an appropriate choice of the number of qubits used by the quantum computer.

4 Conclusion

In summary, the Deutsch-Jozsa, Grover's and Shor's algorithms form examples of how quantum computing is able to provide vast increases in performance over its classical counterpart. This is achieved specifically through the use of quantum properties of matter which are unavailable on classical systems. However, this in turn means that the observed efficiency relies heavily on a fundamental change in the computing approach. In particular, a new quantum paradigm of computation must be adopted, consisting of using the properties of quantum mechanical superposition and interference effectively, and realising that deterministic results may not always be possible. The work of Deutsch, Jozsa, Grover, and Shor exemplifies how this change in algorithm design allows for the improved performance, thereby proving its importance for future development in the field.

References

[1] Ashley Montanaro. "Quantum algorithms: an overview". In: *NPJ Quantum Information* 2.1 (2016), p. 15023. ISSN: 2056-6387.

[2] David Deutsch. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". In: *Proceedings of the Royal Society of London. Series A, Mathematical and physical sciences* 400.1818 (1985), pp. 97–117. ISSN: 1364-5021.

[3] David Deutsch and Richard Jozsa. "Rapid Solution of Problems by Quantum Computation". In: *Proceedings of the Royal Society. A, Mathematical and physical sciences* 439.1907 (1992), pp. 553–558. ISSN: 1364-5021.

[4] R Cleve et al. "Quantum algorithms revisited". In: *Proceedings of the Royal Society. A, Mathematical, physical, and engineering sciences* 454.1969 (1998), pp. 339–354. ISSN: 1364-5021.

[5] David Deutsch. “Quantum theory as a universal physical theory”. In: *International journal of theoretical physics* 24.1 (1985), pp. 1–41. ISSN: 0020-7748.

[6] Lov Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on theory of computing*. STOC ’96. ACM, 1996, pp. 212–219. ISBN: 0897917855.

[7] C. Lavor, L. R. U. Manssur, and R. Portugal. *Grover’s Algorithm: Quantum Database Search*. 2003.

[8] Michel Boyer et al. “Tight Bounds on Quantum Searching”. In: *Fortschritte der Physik* 46.4-5 (June 1998), pp. 493–505.

[9] Peter Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM journal on computing* 26.5 (1997), pp. 1484–1509. ISSN: 0097-5397.