Research Article

Analysis of grover's quantum search algorithm on a classical computer: Identifying opportunities for improvement

Necati ÇELİK[1],*　, Özkan BİNGÖL[2]

[1]Department of Physics Engineering, Gümüşhane University, Gümüşhane, Türkiye
[2]Department of Software Engineering, Gümüşhane University, Gümüşhane, Türkiye

ABSTRACT

In this paper, Grover's quantum search algorithm is analyzed using a classical computer by calculating the amplitudes and the probabilities of finding a single marked state for n=5, 10, 15, 20, 25, and 27 qubit states. The calculations show that the marked state can be found in $\mathcal{O}(\sqrt{N})$ iterations, where $N = 2^n$ is the number of items. The possibility of improving Grover's search algorithm to find a single item in N search elements is discussed by calculating the amplitudes and hence the probabilities of finding a single marked state for n=5, 10, 15, 20, 25, 30, 35, 40, 45, and 50 qubit states. The calculations showed that the marked state could be found with sufficiently high probability in $\mathcal{O}(\ln(N))$ iterations. This is quite a remarkable speed-up that can be achieved to find a single marked element in an unsorted N search element.

Cite this article as: Çelik N, Bingöl Ö. Analysis of grover's quantum search algorithm on a classical computer: Identifying opportunities for improvement. Sigma J Eng Nat Sci 2024;42(4):2039–2049.

INTRODUCTION

Grover's search algorithm [1] provides a relatively fast way of finding a marked element in an unsorted database of size N in the computational complexity $\mathcal{O}(\sqrt{N})$ offered by quantum computers with respect to its classical counterpart, which can only solve the problem in complexity $\mathcal{O}(N)$. This means less memory requirement to represent the input data. The speed-up is a good example of realizing the computational power of quantum computers [2]. This power comes from the laws of quantum mechanics, which are fundamentally different from classical ones. Grover's search algorithm can perform multiple computations simultaneously due to the quantum parallelism. Amplitude of the marked state is amplified at each iteration therefore when measured, the probability of finding the marked element is sufficiently high to be identified. Grover's algorithm is a black-box algorithm, which means that it does not rely on any specific properties of the input data or the function being evaluated. It only requires access to the function that evaluates the input data.

There are numerous useful classical algorithms solving different problems such as generic algorithm to solve second-order boundary value problems [3], for prediction of linear dynamical systems [4], for optimization solution of Troesch's and Bratu's problems [5], and solving singular

Corresponding author.
E-mail address: necati.celik@gumushane.edu.tr

two-point boundary value problems [6], control of axially moving viscoelastic strip [7], adaptive stabilization of a Kirchoff moving strip [8], and existence and stabilization of a Kirchoff moving strip [9]. Apart from classical ones, there are other quantum algorithms, which solve a given problem more efficiently than corresponding classical algorithms: Shor's algorithm [10,11] solves the problem of factorizing large numbers, which is infeasible on a classical computer. The other one, which should be mentioned here, is the Deusch-Jozsa algorithm [12-14] which finds the character of the boolean function exponentially faster than a classical computer.

After being theoretically proposed, Grover's search algorithm has been implemented in many physical systems. These physical systems include nuclear magnetic resonance (NMR) [15, 16], trapped ions [17-20], superconducting qubits in the quantum electrodynamics QED [21-23], and cavity QED [24-26].

Since it is proposed, the mathematical structure of Grover's search algorithm has been investigated [27, 28] and the complexity analysis was studied [29] and applied in many areas such as [30] in which a new protocol for quantum private query is proposed, which achieves efficiency by utilizing two oracle operations to encode both encryption keys and encrypted data items into corresponding quantum superposition states. The protocol also introduces the use of Grover iteration to extract the target from the superposition state. As well known, quantum computers operate upon a register of qubits, which are the quantum mechanical version of classical bits. A qubit can be described as the superposition of 0 and 1, while a classical bit is confined to a definite value of either one of these numbers. That is, describing the state of an n-bit classical information scale as the number of bits. Thus, the calculation time and the memory resources, which are necessary for classical computation, are generally a polynomial function. However, describing the state of n-qubit information requires 2n complex numbers. Therefore, simulating a quantum computer using a classical computer is exponentially complex with respect to the number of qubits. Although classical simulation of quantum computation is exponentially costly, it is vital for the study of new quantum algorithms and architectures. Classical simulation is also important to investigate the efficiencies and robustness of quantum algorithms as compared to their classical counterparts.

To our best knowledge, there is no quantitative analysis of Grover's search algorithm to show how the amplitudes of marked and unmarked states change at every iteration. This is important to see the evolution of the state vector at every iteration and to be able to discuss the possible improvement of the algorithm to obtain a better speed-up. Therefore, the main objective of the current study is to analyze Grover's search algorithm on a classical computer for various numbers of qubits and try to find possible ways to improve it for a better speed-up in finding marked elements in a certain search space.

MATERIALS AND METHODS

Grover's search algorithm proposes to find a particular element through a search space of N elements. Rather than searching for the element directly, one can concentrate on the index to those elements which is a number ranging from 0 to N-1. Defining N=2n, one can store the index in n bits. A particular instance of the search problem can be represented by a function f taking an integer x as an input in the range 0 to N-1. By definition, f(x)=1 if x is a solution to the search problem which is denoted by x*, and if x is not the solution to the search problem, then f(x)=0.

Grover's search algorithm consists of two steps. The first step is preparing the initial qubits in an equal superposition state by applying the Hadamard gates [31]. The second step is applying the Grover iteration $\mathcal{O}(\sqrt{N})$ times. The Grover iteration itself consists of two steps; running the quantum oracle followed by the diffusion operator. After any iteration, while the amplitude of the marked state grows, all other amplitudes shrink [32].

The recognition of the searched element is achieved by making use of an oracle, which is a unitary operator. The action of the oracle is defined in Equation (1).

$$|x\rangle|q\rangle \xrightarrow{Oracle} |x\rangle|q \oplus f(x)\rangle \tag{1}$$

where $|x\rangle$ is index register, $\oplus$ denotes addition module 2, and $|q\rangle$ is called oracle qubit, which is a single qubit and is flipped if f(x)=1, and is unchanged if f(x)=0. It is useful to put the oracle qubit initially in the state $(|0\rangle - |1\rangle)/\sqrt{2}$. If x is not a solution to the search problem, applying the oracle to $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$ does not do anything. But if x is a solution, then $|0\rangle$ and $|1\rangle$ are interchanged after the action of the oracle, giving a final state $-|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$. Thus, the action of the oracle can be summarized in Equation (2).

$$|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \xrightarrow{Oracle} (-1)^{f(x)}|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \tag{2}$$

Schematic circuit for the quantum search algorithm is depicted in Figure 1.

The schematic circuit shown in Figure 1 consists of n qubits prepared in the $|0\rangle$ state and one oracle qubit prepared in the $|1\rangle$ state. Both of these states pass through the Hadamard gate before a series of Grover iteration operators are applied. Then by measuring the first n-qubits the desired result can be obtained. In the following, we give a detailed description of the circuit by writing the state vectors at every step through the circuit. In the circuit, the first input is called the index register which consists of the state of n qubits all prepared in the $|0\rangle$ state. The second input is called the oracle qubit prepared in the $|1\rangle$ state. Therefore $|\psi_0\rangle$ can be written as in Equation (3).

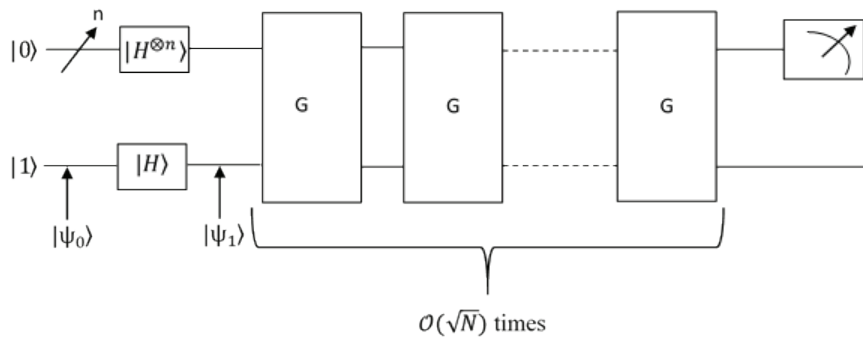$$|\psi_0\rangle = |0^{\otimes n}\rangle|1\rangle \tag{3}$$

Figure 1. Schematic circuit for the quantum search algorithm.

After the Hadamard transform on the index register and the Hadamard transform on the oracle qubit, we have $|\psi_1\rangle$ as given in Equation (4).

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{4}$$

In $|\psi_1\rangle$, the Hadamard transform puts the index register in a superposition of all values which is also called equal superposition state and the oracle qubit is in an evenly weighted superposition of 0 and 1. After that, the circuit consists of repeated application of the Grover iteration which is a quantum subroutine, and applied $\mathcal{O}(\sqrt{2^n})$ times. Figure 2 shows the schematic diagram for the Grover iteration circuit. We can now follow what happens to the state vector from Figure 2.

After the action of the oracle, the state vector takes the form given in Equation (5).

$$|\psi_2\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{5}$$

The action of the oracle is to change the phase of the solution state, which we can also call the marked state $|x^*\rangle$, and separate it from all other states. Thus, a unitary operator as given in Equation (6) can represent the oracle.

$$U_f = I - 2|x^*\rangle\langle x^*| \tag{6}$$

Thus, $|\psi_2\rangle$ can also be written as in Equation (7).

$$|\psi_2\rangle = (I - 2|x^*\rangle\langle x^*|)|S\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{7}$$

where $|S\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n} |x\rangle$ called standard state.

Since the oracle qubit remains $(|0\rangle - |1\rangle)/\sqrt{2}$ throughout the calculations, it can be omitted from further calculations, therefore $|\psi_2\rangle$ can simply be written as in Equation (8).

$$|\psi_2\rangle = (I - 2|x^*\rangle\langle x^*|)|S\rangle \tag{8}$$

After the action of the oracle, the marked state is separated but still, all states have equal amplitudes. Now as the amplitude of the marked state must be increased, while all other amplitudes have to be shrunk so that when measured, the probability of finding the marked state should be the highest. The action of amplitude change can be achieved through the three steps: Hadamard transform
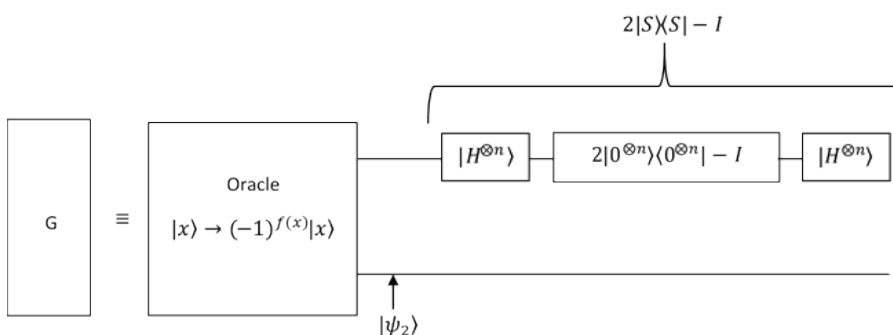

Figure 2. Schematic circuit for the Grover iteration, G.

$H^{\otimes n}$, performing a conditional phase shift with every computational basis except $|0\rangle$ receiving a phase shift of -1, and applying the Hadamard transform $H^{\otimes n}$ again which can be represented by the operator as given in Equation (9) and shown in Figure 2.

$$H^{\otimes n}(2|0^{\otimes n}\rangle\langle 0^{\otimes n}| - I)H^{\otimes n} = 2|S\rangle\langle S| - I = U_s \quad (9)$$

Thus, the Grover iteration may be written as in Equation (10).

$$G = (2|S\rangle\langle S| - I)(I - 2|x^*\rangle\langle x^*|) \quad (10)$$

RESULTS AND DISCUSSION

Analysis of Grover's Search Algorithm

The equal superposition state, $|S\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle$, can be written as the sum of two orthogonal states as given in Equation (11) and shown geometrically in Figure 3.

$$|S\rangle = \frac{\sqrt{N-1}}{\sqrt{N}}|X\rangle + \frac{1}{\sqrt{N}}|x^*\rangle \quad (11)$$

where N is the number of items which is 2n, and n is the number of qubits, $|X\rangle$ and $|x^*\rangle$ are computational basis and $|x^*\rangle$ is marked state and $|X\rangle$ represent all other unmarked states. The evolution of the state vector $|S\rangle$ is calculated by applying Grover's operator G given in Equation (10). Here U_f is called the phase operator which changes the phase of the marked state performed by the oracle and U_s is called

the diffuser which increases the amplitude of the marked state while shrinking the amplitudes of the unmarked states. Probabilities of finding marked and unmarked states are calculated by taking the mod-square of the amplitudes of $|x^*\rangle$ and $|X\rangle$ in Eq. (11), respectively.

The number of iterations has to be known in advance so that when measured, the amplitude of the marked state is sufficiently big and can be distinguished from the unmarked states. As seen from Figure 3, at every iteration, the state vector $|S\rangle$ moves away from the state $|X\rangle$ and approaches $|x^*\rangle$ according to $(2m + 1)\theta$, where m is the number of iterations. Eventually, the angle between $|S\rangle$ and $|X\rangle$ approaches 90 degrees as given in Equation (12).

$$(2m + 1)\theta = \frac{\pi}{2} \quad (12)$$

$m \approx \frac{\pi}{4}\sqrt{N} - \frac{1}{2}$, which is $\mathcal{O}(\sqrt{N})$.

We calculated the evolution of the state vector $|S\rangle$ as given in Equation (13).

$$|\Phi_m\rangle = G|\Phi_{m-1}\rangle \quad ()$$

where $1 \leq m \leq \frac{\pi}{4}\sqrt{N}$ and $|\Phi_0\rangle = |S\rangle = \frac{b}{a}|X\rangle + \frac{1}{a}|x^*\rangle$, $a = \sqrt{N}$ and $b = \sqrt{N-1}$

Some of the calculated state vectors are given in Equations (14-16):

$$|\Phi_1\rangle = \frac{b}{a^3}(2b^2 - 2 - a^2)|X\rangle + \frac{1}{a^3}(2b^2 - 2 + a^2)|x^*\rangle \quad (14)$$

Letting $(2b^2 - 2 - a^2) = c$ and $(2b^2 - 2 + a^2) = d$,

$$|\Phi_2\rangle = \frac{b}{a^5}(2b^2 c - 2d - a^2 c)|X\rangle + \frac{1}{a^5}(2b^2 c - 2d + a^2 c)|x^*\rangle \quad (15)$$

Again letting $(2b^2 c - 2d - a^2 c) = e$ and $(2b^2 c - 2d + a^2 c) = f$

$$|\Phi_3\rangle = \frac{b}{a^7}(2b^2 e - 2f - a^2 e)|X\rangle + \frac{1}{a^7}(2b^2 e - 2f + a^2 f)|x^*\rangle \quad (16)$$

Considering all these steps, the general formula that calculates each $|\Phi_{i+1}\rangle$ vector iteratively is given in Equation (17).

$$|\Phi_{i+1}\rangle = \frac{b}{a^{(2i+1)}}c_i|X\rangle + \frac{1}{a^{(2i+1)}}d_i|x^*\rangle \quad (17)$$

Here, c_i and d_i are the generalizations of all c and d values.

$$c_{i+1} = 2b^2 c_i - 2d_i - a^2 c_i \quad (18)$$

$$d_{i+1} \leftarrow 2b^2 c_i - 2d_i - a^2 d_i \quad (19)$$

Figure 3. Evolution of equal superposition state $|S\rangle$ in Grover's method.

To calculate the amplitudes, a function suitable for the pseudo-code given in Algorithm-1 was created.

Algorithm 1. Pseudo Code of Grover's Search Algorithm

Inputs : n – number of qubits

n_step – number of iterations

Outputs : A[] – amplitudes of unmarked states

B[] – amplitudes of marked state

Begin

Set $N \leftarrow 2^n$, % number of search space

Set $a \leftarrow \sqrt{N}$

Set $b \leftarrow \sqrt{N-1}$

Set $c_0 \leftarrow 1$

Set $d_0 \leftarrow 1$

Set $i \leftarrow 0$

while $i < n\_step$ **do**

$A_i \leftarrow \frac{b.c_i}{a^{(2i+1)}}$

$B_i \leftarrow \frac{d_i}{a^{(2i+1)}}$

$|\Phi_{i+1}\rangle \leftarrow A_i|X\rangle + B_i|x^*\rangle$

$c_{i+1} \leftarrow 2b^2 c_i - 2d_i - a^2 c_i$

$d_{i+1} \leftarrow 2b^2 c_i - 2d_i - a^2 d_i$

$i \leftarrow i + 1$

end while

End

Depending on the number of qubits, a maximum number of 10000 iterations were analyzed and the amplitudes (and probabilities) were calculated and shown in Figure (a-f) 4 for n=5, 10, 15, 20, 25, and 27 qubits. For n=2 qubit state (N=2²=4 search space), only one iteration is required to find the marked state with probability one (not shown in Figure 4).

As the number of qubits increases, the number of iterations also increases smoothly according to $\sim \frac{\pi}{4}\sqrt{N}$. Figure 4 shows that as the number of iterations increases, while the probability of finding the marked state increases, the probability of finding the unmarked states decreases. The number of iterations required to find the marked state with sufficiently high probability for n=5, 10, 15, 20, 15, and 27 qubits is 4, 25, 142, 804, 4549, and 9099 respectively as shown in Table 1.

The Analysis of Proposed Method

The equal superposition state can be given in Equation (20).

$$|S\rangle = \frac{b}{a}|X\rangle + \frac{1}{a}|x^*\rangle \qquad (20)$$

where, $a = \sqrt{N}$ and $b = \sqrt{N-1}$

The phase-changing operator is the same as in the case of Grover's method and given as $U_f = I - 2|x^*\rangle\langle x^*|$. In the proposed method, as different from Grover's method, the diffuser operator changes in every iteration, which is given in Equation (21).

$$U_{s_m} = 2|\Phi_m\rangle\langle\Phi_m| - I \qquad (21)$$

where $|\Phi_m\rangle$ is the new state vector after $G_m = U_{s_m} U_s$ operator acts, where $1 \leq m \leq \mathcal{O}(\ln(N))$. After each Grover operator, the change in the state vectors is shown in Figure 5.

Note that the main difference between Grover's method and the proposed method is that the diffuser operator changes at every iteration. In Grover's method, the diffuser operator is $2|S\rangle\langle S| - I$ and stays unchanged throughout the search process. In the proposed method, however it consists of the final state vector at the end of each iteration and the unit operator. Some of the state vectors after the action of Grover's operator are given in Equations (22-24).

$$|\Phi_1\rangle = \frac{b}{a^3}(2b^2 - 2 - a^2)|X\rangle + \frac{1}{a^3}(2b^2 - 2 + a^2)|x^*\rangle \qquad (22)$$

Letting $(2b^2 - 2 - a^2) = c_1$ and $(2b^2 - 2 + a^2) = d_1$,

$$|\Phi_2\rangle = \frac{bc}{a^9}(2b^2 c_1^2 - 2d_1^2 - a^6)|X\rangle + \frac{1}{a^9}(2b^2 c_1^2 - 2d_1^2 + a^6)|x^*\rangle \qquad (23)$$

Again letting $(2b^2 c_1^2 - 2d_1^2 - a^6) = c_2$ and $(2b^2 c_1^2 - 2d_1^2 + a^6) = d_2$

$$|\Phi_3\rangle = \frac{bc_1c_2}{a^{27}}(2b^2 c_1^2 c_2^2 - 2d_1^2 d_2^2 - a^{18})|X\rangle + \frac{1}{a^{27}}(2b^2 c_1^2 c_2^2 - 2d_1^2 d_2^2 + a^{18})|x^*\rangle \qquad (24)$$

The general formula that calculates each $|\Phi_{i+1}\rangle$ vector iteratively is given in Equation (25).

$$|\Phi_{i+1}\rangle = \frac{b}{a^{3^{i+1}}}\left(\prod_{j=1}^{i} c_j\right)\left(T_{i+1} - a^{2.3^i}\right)|X\rangle + \frac{1}{a^{3^{i+1}}}\left(T_{i+1} + a^{2.3^i}\right)|x^*\rangle \qquad (25)$$

Here T_{i+1} represents a common operation and is specified in Equation (26).

$$|T_{i+1}\rangle = 2b^2\prod_{j=1}^{i} c_j^2 - 2\prod_{j=1}^{i} d_j^2 \qquad (26)$$

Similarly, ci and di are generalized values of all c and d and can be formulated as

$$c_{i+1} \leftarrow 2b^2 c_i^2 - 2d_i^2 - a^{2.3^i} \qquad (27)$$

$$d_{i+1} \leftarrow 2b^2 c_i^2 - 2d_i^2 + a^{2.3^i} \qquad (28)$$

To calculate the amplitude in the standard state of the proposed method, a piece of code is written as in Algorithm 2.

As a characteristic future of Grover's search algorithm, the number of iterations has to be known prior to the
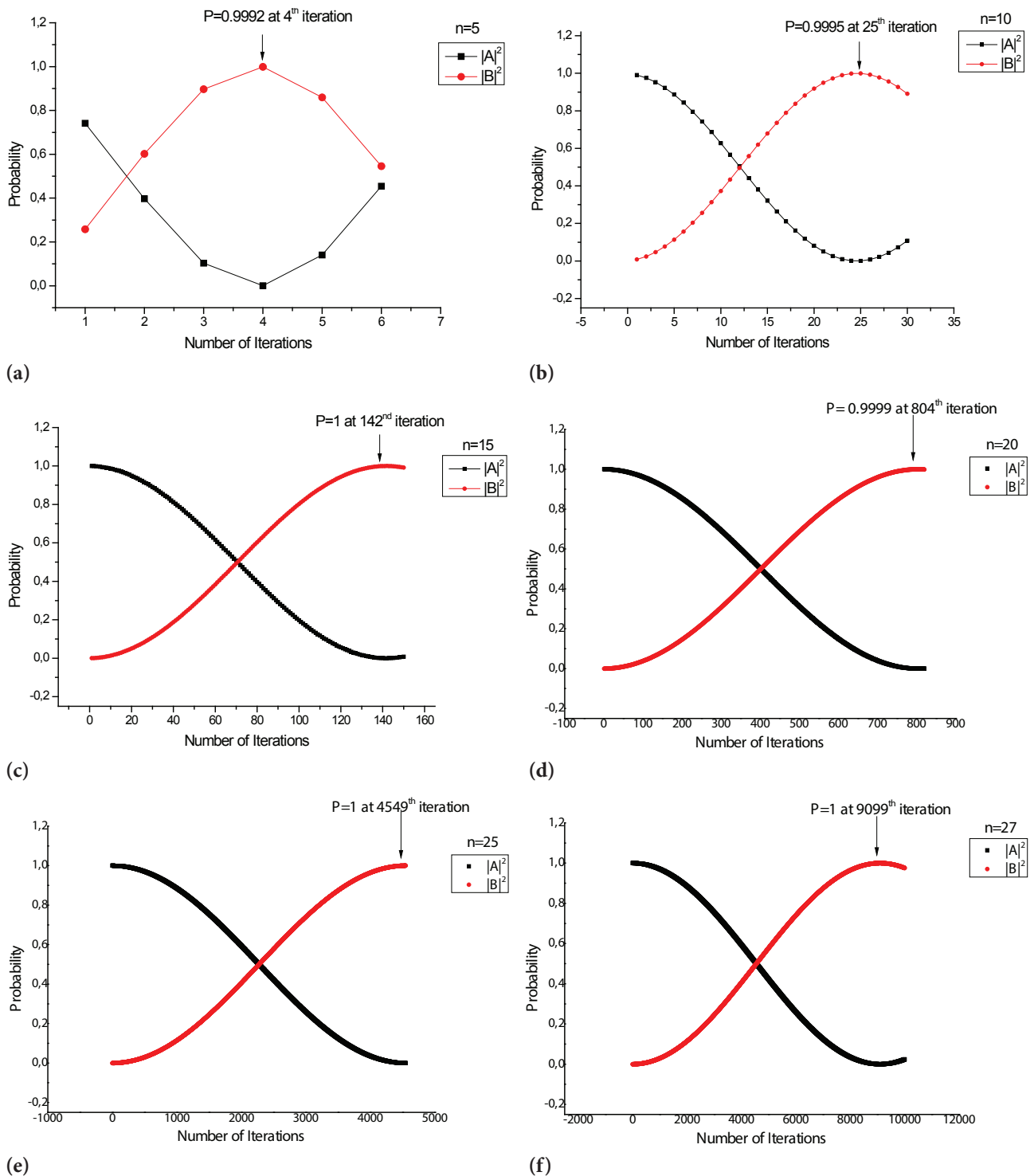
Figure 4. Probabilities finding the marked state and unmarked states.

calculations. The complexity of the proposed method is analyzed as follows. As seen in Figure 5, in every iteration, the state vector approaches the marked state according to Equation (29).

$$3^m \theta \approx \frac{\pi}{2} \tag{29}$$

where m is the estimated number of iterations and given in Equations (30, 31).

$$m \approx \frac{\ln\left(\frac{\pi}{2}\right) + \frac{1}{2}\ln(N)}{ln3} \tag{30}$$
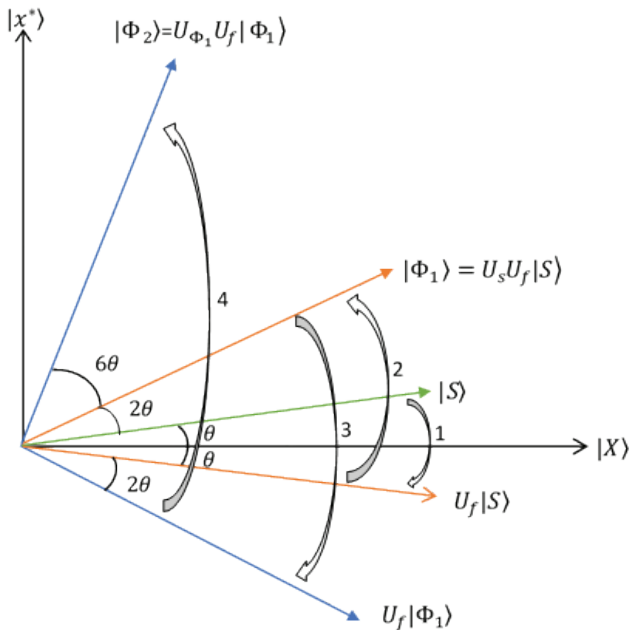
$$m = \mathcal{O}(lnN) \tag{31}$$

Figure 5. Evolution of equal superposition state $|S\rangle$ in proposed method.

Algorithm 2. Pseudo Code of Proposed Method

|	
Inputs	: n – number of qubits
	n_step – number of iterations
Outputs	: A[] – amplitudes of unmarked states
	B[] – amplitudes of marked state

Begin

    $N \leftarrow 2^n$, % number of search space

    Set $a \leftarrow \sqrt{N}$

    Set $b \leftarrow \sqrt{N-1}$

    Set $c_0 \leftarrow 1$

    Set $d_0 \leftarrow 1$

    Set $i \leftarrow 0$

    while $i < n\_step$ **do**

        Set $prod\_c_j \leftarrow 1$, % $prod\_c_j$ is $\prod_{j=1}^{i} c_j$

        Set $prod\_c_j^2 \leftarrow 1$, % $prod\_c_j^2$ is $\prod_{j=1}^{i} c_j^2$

        Set $prod\_d_j^2 \leftarrow 1$, % $prod\_d_j^2$ is $\prod_{j=1}^{i} d_j^2$

        Set $j \leftarrow 0$

        while $j \le i$ **do**

            $prod\_c_j \leftarrow prod\_c_j * c_j$

            $prod\_c_j^2 \leftarrow prod\_c_j^2 * c_j$

            $prod\_d_j^2 \leftarrow prod\_d_j^2 * d_j * d_j$

            $j \leftarrow j + 1$

        end while

        $T_{i+1} \leftarrow 2b^2 prod\_c_j^2 - 2prod\_d_j^2$

        $A_i \leftarrow \dfrac{b.prod\_c_j}{a^{3^{i+1}}}\left(T_{i+1} - a^{2.3^i}\right)$

        $B_i \leftarrow \dfrac{1}{a^{3^{i+1}}}\left(T_{i+1} + a^{2.3^i}\right)$

        $|\Phi_{i+1}\rangle \leftarrow A_i|X\rangle + B_i|x^*\rangle$

        $c_{i+1} \leftarrow 2b^2 c_i^2 - 2d_i^2 - a^{2.3^i}$

        $d_{i+1} \leftarrow 2b^2 c_i^2 - 2d_i^2 + a^{2.3^i}$

        $i \leftarrow i + 1$

    end while

End

As seen, an exponential speed-up is achieved in the proposed method. At every iteration, the amplitudes (and hence the probabilities) of unmarked and marked states were calculated. The calculations were performed for n=5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 qubits. The results are shown in Figure 6 (a-j).

As for the case of n=2, only one iteration is required to find the marked state with probability one (not shown in Figure 6). The calculations showed that for n= 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50 qubits, the number of iterations required to find the marked state with relatively higher probability as compared to the unmarked states are 2, 5, 5, 7, 8, 10, 11, 13, 16, and 16, respectively as shown in Table 1.

When compared to Grover's method, in some cases, the probability of finding the marked state is relatively low in the proposed method. For example, for the cases n=20, 35 and 45 qubits, the probabilities of finding the marked states are about 0.7. For all other cases, the probabilities are about 0.9, which is sufficiently high.

The implementation of the methods mentioned in this study was carried out on a PC with an i5-8400 processor and 8 GB memory with Windows 10 operating system. MATLAB [33] was preferred as the software development environment. HPF Toolbox [34] is used to solve the large number of problems encountered due to the high number of exponential and multiplication operations during coding.

As it is clearly seen in Eq. (27-28), c and d values are calculated iteratively and each value takes the value calculated in the previous iteration as a multiplier. However, the a^{3^i} value is calculated at each stage. Here i denotes the iteration number. Therefore, as the iteration number increases, the value of a^{3^i} grows very quickly. Since the classical computer architecture does not allow the calculation of such large numbers, the HPF [34] library was used. With this library, instead of allocating a standard 16-digit memory for a variable, higher numbers of digits can be used. We preferred variable lengths of 10-200-500-1000 and 2000 digits for the very large numbers that emerged in this study. Although the use of more digits allows much larger numbers to be calculated, we had to terminate it at some point because the calculation time was prolonged. The average of the times required to calculate all other values in an iteration is given in the last column of the table. It is clearly seen in the table that the calculated values and times grow exponentially. However, especially after the 32nd iteration (usually around the 40th iteration), a buffer overflow occurs in the calculated values for all qubits. When all these situations are
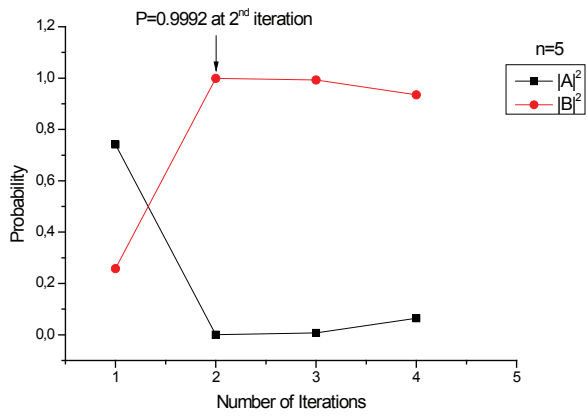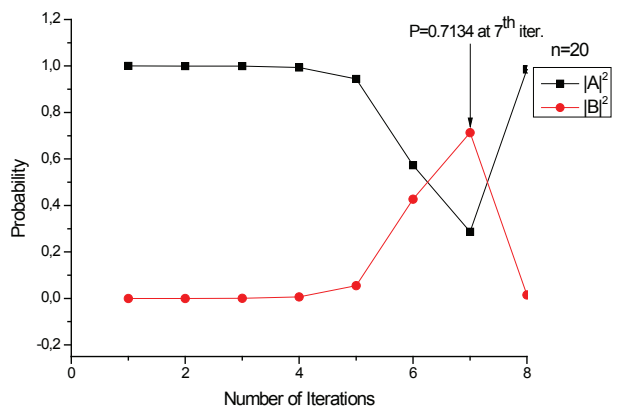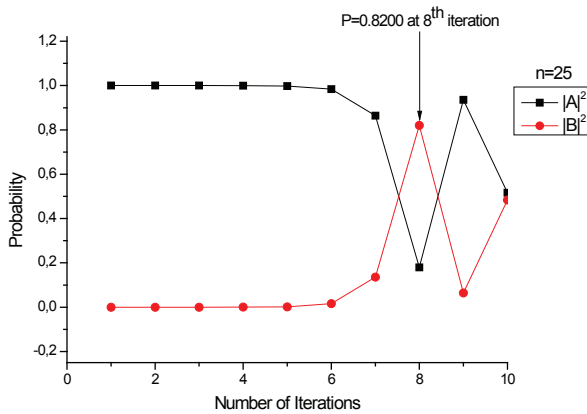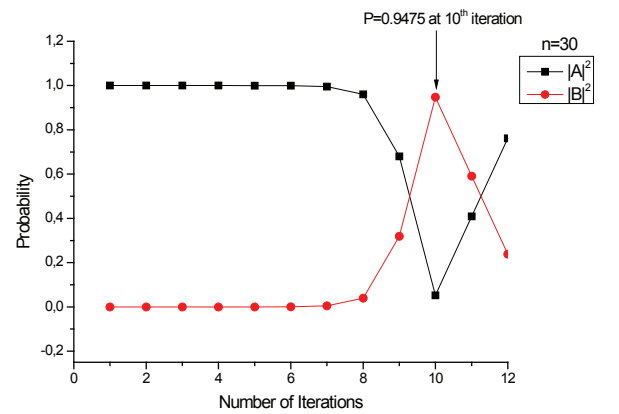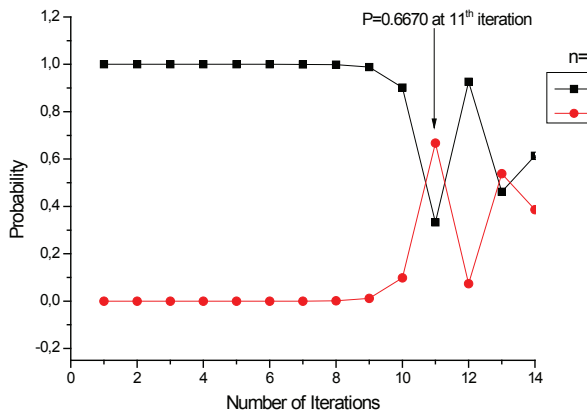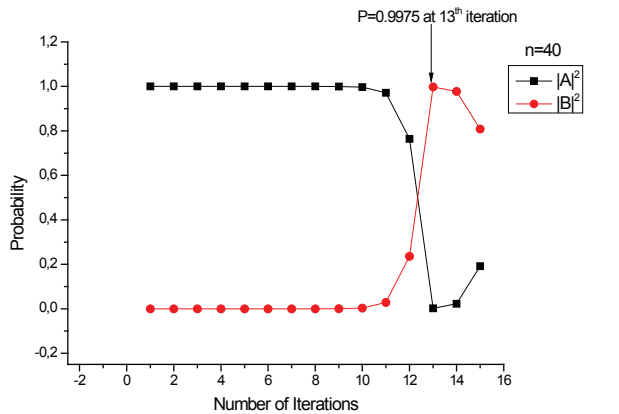
(a)
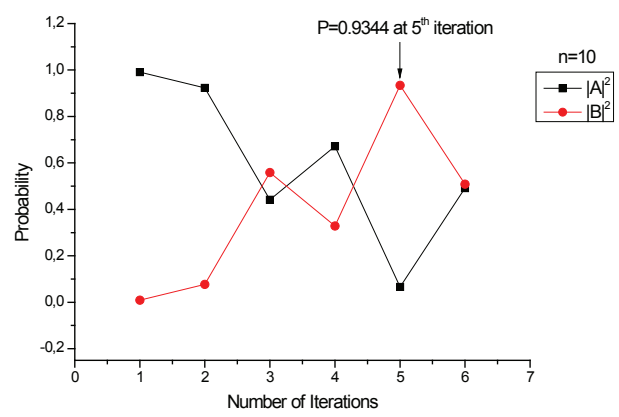
(b)
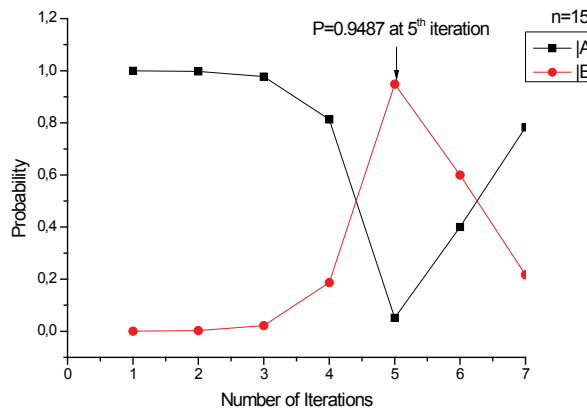
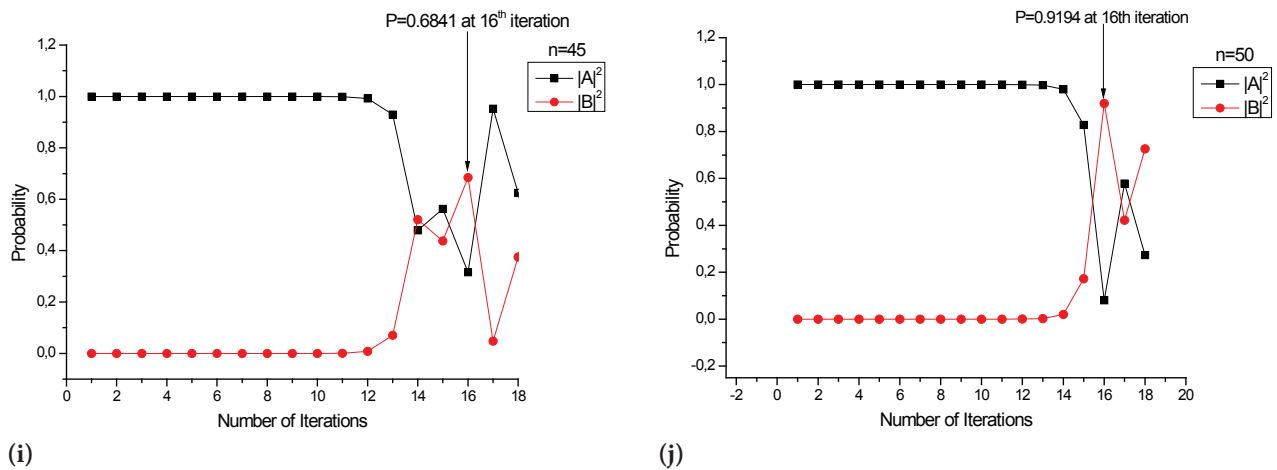(c)

(d)

(e)

(f)

(g)

(h)

Figure 6. Probabilities of finding the marked and unmarked states.

Table 1. Comparison of the number of iterations and elapsed times for Grover's and proposed method

		Number of Iterations and Elapsed Time			
Number of qubits (n)	Number of search space (N)	Iteration Number (Grover's Method)	Elapsed Time (Grover's Method)	Iteration Number (Proposed Method)	Elapsed Time (Proposed Method)
1	2	1	0.2 s	1	0.04 s
5	32	4	0.8 s	2	0.07 s
10	1024	25	5 s	5	0.18 s
15	32768	142	28.4 s	5	0.18 s
20	1048576	804	160.8 s	7	0.24 s
25	33554432	4549	909.8 s	8	0.29 s
27	134217728	9099	1819.8 s	9	0.32 s
30	1073741824	25736	5547.2 s	10	0.35 s
35	34359738368	-		11	0.40 s
40	1.09951E+12	-		13	0.46 s
45	3.51844E+13	-		16	0.56 s
50	1.1259E+15	-		16	0.56 s

evaluated, libraries are needed in which much larger numbers can be used in order to reach 1000 iterations in total. The proposed approach was coded with the MATLAB program, and a more efficient library than HPF was not found. In this regard, it is planned to examine the libraries used for finding large prime numbers and to develop the developed approach with these libraries.

CONCLUSION

To our knowledge, there is no quantitative analysis of Grover's iterative search algorithm. In the current paper, we showed how the amplitudes and hence the probabilities of finding a single marked state and unmarked states

evolve at each iteration for various qubit states. We also discussed the potential improvement of Grover's quantum search algorithm. The study performed in the current study can be summarized in two steps: First, we analyzed the quantitative analysis of Grover's search algorithm by calculating the amplitudes and hence the probabilities of finding the marked and unmarked states for various qubit states from n=5 to n=27 qubits. The calculations showed that while the probabilities of the marked states increase smoothly with increasing the number of iterations, the probabilities of unmarked states decrease smoothly with increasing the number of iterations. Second, we proposed an improved version of Grover's search algorithm. The calculations showed that a great speed-up is achieved after

improvement is made to Grover's search algorithm. For example, suppose a marked element is desired to be found in N=106 elements. Grover search algorithm can find that element in $\sqrt{10^6} = 10^3 = 1000$ iterations. However, after the improvement one can find the same element in only $\ln(10^6) \approx 14$ iterations which is a great speed up in the search algorithm. However, in the proposed method, for some cases, the probability of finding the marked state does not smoothly increase, and for some cases, the probability of finding the marked state is not high enough. That would limit the use of our proposed method in real applications.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Grover LK. Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett 1997;79:325. [CrosssRef]

[2] Biham E, Biham O, Biron D, Grassl M, Lidar DA, Shapira D. Analysis of generalized Grover quantum search algorithms using recursion equations. Phys Rev A 2000;63:012310. [CrosssRef]

[3] Arqub OA, Abo-Hammour Z. Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. Inf Sci 2014;279:396–415. [CrosssRef]

[4] Abo-Hammour ZE, Alsmadi O, Momani S, Abu Arqub O. A genetic algorithm approach for prediction of linear dynamical systems. Math Probl Eng 2013;2013:831657. [CrosssRef]

[5] Abo-Hammour Z, Abu Arqub O, Momani S, Shawagfeh N. Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm. Discrete Dyn Nat Soc 2014;2014:401696. [CrosssRef]

[6] Abu Arqub O, Abo-Hammour Z, Momani S, Shawagfeh N. Solving singular two-point boundary value problems using continuous genetic algorithm. Abstr Appl Anal 2012;2012:205391. [CrosssRef]

[7] Kelleche A, Tatar NE. Control and exponential stabilization for the equation of an axially moving viscoelastic strip. Math Meth Appl Sci 2017;40:6239–6253. [CrosssRef]

[8] Kelleche A, Tatar NE. Adaptive Stabilization of a Kirchhoff moving string. J Dyn Control Syst 2020;26:255–263. [CrosssRef]

[9] Kelleche A, Tatar NE. Existence and stabilization of a Kirchhoff moving string with a distributed delay in the boundary feedback. Math Model Nat Phenom 2017;12:106–117. [CrosssRef]

[10] Kaye P, Laflamme R, Mosca M. An Introduction to Quantum Computing. Oxford, London: OUP; 2006. [CrosssRef]

[11] Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev 1999;41:303–332. [CrosssRef]

[12] Qiu D, Zheng S. Revisiting Deutsch-Jozsa algorithm. Inf Comput. 2020;275:104605. [CrosssRef]

[13] Deutsch D, Jozsa R. Rapid solution of problems by quantum computation. Proc R Soc Lond A Math Phys Sci 1992;439:553–558. [CrosssRef]

[14] Kiktenko EO, Fedorov AK, Strakhov AA, Man'Ko VI. Single qudit realization of the Deutsch algorithm using superconducting many-level quantum circuits. Phys Lett A 2015;379:1409–1413. [CrosssRef]

[15] Chuang IL, Gershenfeld N, Kubinec M. Experimental implementation of fast quantum searching. Phys Rev Lett 1998;80:3408. [CrosssRef]

[16] Chuang IL, Vandersypen LM, Zhou X, Leung DW, Lloyd S. Experimental realization of a quantum algorithm. Nature 1998;393:143–146. [CrosssRef]

[17] Feng M. Grover search with pairs of trapped ions. Phys Rev A 2001;63:052308. [CrosssRef]

[18] Brickman KA, Haljan PC, Lee PJ, Acton M, Deslauriers L, Monroe C. Implementation of Grover's quantum search algorithm in a scalable system. Phys Rev A 2005;72:050306. [CrosssRef]

[19] Xu ZY, Feng M. Addendum to "Grover search with pairs of trapped ions". Phys Rev A 2008;78:014301. [CrosssRef]

[20] Yang WL, Wei H, Zhou F, Chang WL, Feng M. Solution to the satisfiability problem using a complete Grover search with trapped ions. J Phys B At Mol Opt Phys 2009;42:145503. [CrosssRef]

[21] Dewes A, Lauro R, Ong FR, Schmitt V, Milman P, Bertet P, et al. Quantum speeding-up of computation demonstrated in a superconducting two-qubit processor. Phys Rev B 2012;85:140503. [CrosssRef]

[22] Dewes A, Ong FR, Schmitt V, Lauro R, Boulant N, Bertet P, et al. Characterization of a two-transmon processor with individual single-shot qubit readout. Phys Rev Lett 2012;108:057002. [CrosssRef]

[23] Filipp S, Maurer P, Leek PJ, Baur M, Bianchetti R, Fink JM, et al. Two-qubit state tomography using a joint dispersive readout. Phys Rev Lett 2009;102:200402. [CrosssRef]

[24] Yamaguchi F, Milman P, Brune M, Raimond JM, Haroche S. Quantum search with two-atom collisions in cavity QED. Phys Rev A 2002;66:010302. [CrosssRef]

[25] Wang H, Zhang S, Yeon KH. Implementation of Grover quantum search via cavity quantum electrodynamics. J Korean Phys Soc 2008;53:3144–150. [CrosssRef]

[26] Yang WL, Chen CY, Feng M. Implementation of three-qubit Grover search in cavity quantum electrodynamics. Phys Rev A 2007;76:054301. [CrosssRef]

[27] Hua M, Tao MJ, Deng FG. Fast universal quantum gates on microwave photons with all-resonance operations in circuit QED. Sci Rep 2015;5:1–8. [CrosssRef]

[28] Szabłowski PJ. Understanding mathematics of Grover's algorithm. Quantum Inf Process 2021;20:1–21. [CrosssRef]

[29] Han Q. Several remarks on Grover's quantum search algorithm with a single marked element. Sch J Phys Math Stat 2021;3:62–67. [CrosssRef]

[30] Cohn I, De Oliveira ALF, Buksman E, De Lacalle JG. Grover's search with local and total depolarizing channel errors: Complexity analysis. Int J Quantum Inf 2016;14:1650009. [CrosssRef]

[31] Xiao H, Huang WH, Zhou M. An efficient quantum private query protocol based on Oracle and Grover iteration. Int J Theor Phys 2019;58:3025–3035. [CrosssRef]

[32] Nielsen MA, Chuang I. Quantum Computation and Quantum İnformation. Cambridge: Cambridge University Press; 2002.

[33] Get a Trial of MATLAB and Simulink Product. Accessed onJul 12, 2024. Available at: https://uk.mathworks.com/

[34] D'Errico J. HPF - a big decimal class. MATLAB Central File Exchange. Accessed on Jan 10, 2022. Available at: https://www.mathworks.com/matlabcentral/fileexchange/36534-hpf-a-big-decimal-class