Scientific Research Publishing

American Journal of Computational Mathematics, 2024, 14, 169-188
https://www.scirp.org/journal/ajcm
ISSN Online: 2161-1211
ISSN Print: 2161-1203

O(log*N*) Algorithm for Amplitude Amplification and O(log*N*) Algorithms for Amplitude Transfer in Grover's Algorithm

Ying Liu

Department of Engineering Technology, Savannah State University, Savannah, Georgia, USA
Email: liuy@savannahstate.edu

How to cite this paper: Liu, Y. (2024) O(log*N*) Algorithm for Amplitude Amplification and O(log*N*) Algorithms for Amplitude Transfer in Grover's Algorithm. *American Journal of Computational Mathematics*, **14**, 169-188.
https://doi.org/10.4236/ajcm.2024.142005

Received: March 13, 2024
Accepted: May 5, 2024
Published: May 8, 2024

Abstract

Grover's algorithm is a category of quantum algorithms that can be applied to many problems through the exploitation of quantum parallelism. The Amplitude Amplification in Grover's algorithm is $T = O(\sqrt{N})$. This paper introduces two new algorithms for Amplitude Amplification in Grover's algorithm with a time complexity of $T = O(\log N)$, aiming to improve efficiency in quantum computing. The difference between Grover's algorithm and our first algorithm is that the Amplitude Amplification ratio in Grover's algorithm is an arithmetic series and ours, a geometric one. Because our Amplitude Amplification ratios converge much faster, the time complexity is improved significantly. In our second algorithm, we introduced a new concept, Amplitude Transfer where the marked state is transferred to a new set of qubits such that the new qubit state is an eigenstate of measurable variables. When the new qubit quantum state is measured, with high probability, the correct solution will be obtained.

Keywords

Quantum Computing, Oracle, Amplitude Amplification, Grover's Algorithm

1. Introduction

Quantum Computing [1] [2] [3] [4] [5] is a field of computing that leverages the principles of Quantum Mechanics. Quantum Computing has strange phenomena known as Superposition and Entanglement. Grover's algorithm [6] is one of the most famous quantum algorithms that provide a quadratic speedup over the best classical algorithms for unstructured search problems, *i.e.* from $T = O(N)$ to $T = O(\sqrt{N})$. It was proposed by Lov Grover [6] in 1996 and is a fundamental

algorithm in the field of Quantum Computing. Its efficiency arises from the exploitation of quantum parallelism and quantum interference. Furthermore, it has evolved into a category of algorithms that can be applied to many problems, such as SAT [7], and Subset Sum [8].

Grover's algorithm [6] is:

Initialization;
Oracle;
for ($i = 0$; $i < \mathrm{O}(\sqrt{N})$, i++)
    Amplitude Amplification;
Measurement;

Where:

- Initialization: Start with a superposition of all possible states. If there are $N = 2^n$ possible solutions, where n is the number of qubits, this superposition is created over N states.
- Oracle: Introduce an Oracle gate that identifies the target solution.
- Amplitude Amplification: Apply a series of quantum operations that amplify the amplitude of the marked state and suppress the amplitudes of the other states.
- Repeat Amplification: Amplifications are repeated for a certain number of iterations.
- Measurement: The quantum state is measured, and with high probability, the correct solution is obtained.

In Grover's algorithm, the key is Oracle and Amplitude Amplification. The job of the Oracle is to mark the solution. The job of Amplitude Amplification is that the amplitudes of incorrect states experience destructive interference, reducing their probabilities, while the amplitude of the correct state experiences constructive interference, increasing its probability.

This paper introduces two new algorithms for Amplitude Amplification in Grover's algorithm with a time complexity of $T = \mathrm{O}(\log N)$, aiming to improve efficiency in quantum computing.

In the first algorithm, we will show that through rotation in each amplitude amplification, the ratio between the amplitude of the marked state and other states forms a geometric series, thus achieving the logarithmic time steps to reach the threshold for the algorithm to stop: $T = \log(N) = \log(2^n) = \mathrm{O}(n)$, where n is the number of qubits and N is the length of an unsorted list. The difference between the Grover's algorithm and our first algorithm is that the Amplitude Amplification ratio in Grover's algorithm is an arithmetic series and ours, is a geometric one. This difference determines the difference between a square root time complexity and a logarithmic time complexity. Because our Amplitude Amplification ratios converge much faster, the time complexity is improved significantly.

In the second algorithm, we will introduce a new concept, Amplitude Transfer where the marked state is transferred to a new set of qubits such that the new

qubit state is an eigenstate of measurable variables. When the new qubit quantum state is measured, with high probability, the correct solution will be obtained.

Section 2. Basic Notation' first introduces $X = \{0, 1\}^d$ space, then, we will introduce the notation for superposition vector.

Section 3. The Well-known Oracle Assumption' describes the well-known Oracle assumption.

Section 4. O(log$N$) Algorithm for Amplitude Amplification' introduces the first O(log$N$) algorithm. In particular, the difference between Grover's algorithm and our first algorithm is that the Amplitude Amplification ratio in Grover's algorithm is an arithmetic series and ours, a geometric one. The underlying methodology of this algorithm is that our geometric series stops much faster than Grover's arithmetic series. Because our Amplitude Amplification ratios converge much faster, the time complexity is improved significantly.

Section 5. Visualizations' informally prove that the rotation matrix for the rotation introduced in Section 4 is unitary.

Section 6. Amplitude Transfer' introduces a new concept, Amplitude Transfer where the marked state is transferred to a new set of qubits such that the new qubit state is an eigenstate of measurable variables. When the new qubit quantum state is measured, with high probability, the correct solution will be obtained.

Section 7. The Gate Equation Used' lists the well-known gate equations used in the second O(log$N$) algorithm.

Section 8. The O(log$N$) Circuit Design' will construct a quantum circuit for the Amplitude Amplification in Grover's algorithm such that the Time Complexity is: T = O(log$N$). We will use a simple example to show the design and then, generalize the design.

Section 9. Modified Unstructured Search' presents a O(log$N$) algorithm for unstructured search.

Section 10. Discussions' discuss the potential applications.

Appendix A. Geometric Series vs Arithmetic Series Comparison' will provide a more thorough comparison with Grover's algorithm. This highlights the underlying methodology: our geometric series stops much faster than Grover's arithmetic series.

2. Basic Notation

Throughout this paper,

n is the number of qubits;

$N = 2^n$ is the number of states;

$L <= N$ is the number of items in an unsorted list;

M is the unsorted list.

We will use the standard notations, meaning we will overload the symbol, X, for both the Instance Space name and Gate name. An Instance Space, $X = \{0, 1\}^n$,

is a set of all instances given in Equation (1):

$$X = \{0\cdots00, 0\cdots01, 0\cdots10, 0\cdots11, \cdots\} \tag{1}$$

An instance of X is $x \in X$, where $x = 00\cdots0$, or, $0\cdots01$, $\cdots$, and $|X| = 2^n$. An instance, x, is a binary string, which can be converted into a decimal number:

$$X = \{0,1,2,3,\cdots,N-1\} \tag{2}$$

An instance, x, can be the qubits. It can appear as a binary string or a decimal number. Examples are: $|000\rangle = |0\rangle$, $|001\rangle = |1\rangle$, …, $|111\rangle = |7\rangle$.

In Grover's algorithm, the starting superposition of all possible states,

$$|\psi\rangle = \sum_{x=0}^{N-1} a(x)|x\rangle \tag{3}$$

is always:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{4}$$

In this paper, k represents the steps: $k = 0, 1, 2, 3, \ldots, K$. For $k = 0$, the starting superposition is Equation (4). Here $a(x)$ is the amplitude of the state at $k = 0$. For $k = 1$,

$$|\psi_1\rangle = \sum_{x=0}^{N-1} a(x, k=1)|x\rangle \tag{5}$$

Here, $a(x, k = 1)$ are the amplitudes at $k = 1$. This iteration in Grover's algorithm will stop at some step, K, when some threshold is reached:

$$|\psi_K\rangle = \sum_{x=0}^{N-1} a(x, K)|x\rangle \tag{6}$$

And the time complexity of the Grover's algorithm is O(K).

Without loss of generality, it is assumed that Oracle will always pick state $x = 1$, $|x=1\rangle$, with amplitude: $a(x = 1, k)$. If the Oracle would pick up an arbitrary index, j, simply replace the discussion below by index j.

We will also omit k whenever possible. The N states are divided into two groups: the marked state and all other states. With exception of the marked state, the amplitudes for all other states are the same for {0, 2, 3, …, $N - 1$}. For this reason, throughout this paper, we will only discuss two amplitudes, $a_0(k) = a(x = 0, k)$ and $a_1(k) = a(x = 1, k)$. This is because with exception of the marked state, all of the other amplitudes are equal:

$$a_0 = a_2 = a_3 = \cdots \tag{7}$$

The threshold for our algorithm to stop is the same as Grover's algorithm, *i.e.* the amplitude of the marked state ≥ $1/\sqrt{2}$ and the probability ≥ 1/2.

3. The Well-Known Oracle Assumption

Grover's paper [6] is to search through the unsorted list and amplify the probability of finding the target state(s) upon measurement. In this particular example, the Oracle's role is to mark the target number which is given. For example,

let $n = 3$, and we want to search an arbitrary target, $|5\rangle = |101\rangle$, from an arbitrary random list, $M = \{3, 4, 6, 0, 1, 2, 7, 5\}$. Since we know $|101\rangle$ is the solution, the Oracle simply marks the |101> state.

In Grover's algorithm, the starting superposition of all possible states is Equation (5). If Equation (5) is measured, all states are equally likely to appear with probability, $P(x) = 1/N$. Oracle circuit checks for some conditions: $f(x) = 0$. Initially, the input to the Oracle is Equation (5) and the output is 00…0.

If a particular state, $x = o$, satisfies some conditions: $f(o) = 0$, this state can pass the Oracle; otherwise, the output is 00…0. This passed state is the marked state. Even if it is marked, the marked state is still entangled with the other state,

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{8}$$

So if the quantum state is measured, the probability for this marked state to appear would still be $1/N$. In other words, the black box can identify the solution(s) without explicitly knowing what they are because of the low probability of $P(x) = 1/N$ for the marked state. Therefore, the Oracle's goal is not to reveal the solution but rather to provide a quantum operation that efficiently marks the solution state(s) within the superposition of states.

In general, the Oracle doesn't "know" the solution in the classical sense. Instead, it's provided with a black-box function that can identify the solution(s) without explicitly knowing what they are. This black box evaluates whether a given state represents a solution and marks it accordingly.

4. O(log*N*) Algorithm for Amplitude Amplification

The difference between the Grover's algorithm and our first algorithm is that the Amplitude Amplification ratio in Grover's algorithm is an arithmetic series, see Appendix A, and ours, a geometric one. Because our Amplitude Amplification ratios converge much faster, the time complexity is improved significantly. The geometric series will result in O(log*N*) algorithm.

This section is organized as follows: Theorem 1 gives the formula for Amplitude Amplification ratio under the geometric series. Theorem 2 shows O(log*N*) Time Complexity under the geometric series. Theorem 3 in the next section is a requirement from Quantum Mechanics. Again, we will only discuss two amplitudes, $a_0(k) = a(x = 0, k)$ and $a_1(k) = a(x = 1, k)$.

Theorem 1. Let $k = 0$ be the initial step with n-qubits given by Equation (5), let $\beta > 1$ be a free parameter, let $k = 0, 1, 2, \ldots, K$ be the iterations in Grover's algorithm, and let r be the Amplitude Amplification ratio between the marked state and other states:

$$r(k) = \frac{a_1(k)}{a_0(k)} \tag{9}$$

If $r(k)$, $k = 0, 1, 2, \ldots, K$, is a geometric series:

$$\beta^0, \beta^1, \beta^2, \cdots, \beta^K, \ \beta > 1, \tag{10}$$

and

$$a_0 = a_2 = a_3 = \cdots$$

then

$$a_1(k) = \frac{\beta^k}{\sqrt{2^n + \beta^{2k} - 1}} \tag{11}$$

$$a_0(k) = \frac{1}{\sqrt{2^n + \beta^{2k} - 1}} \tag{12}$$

Remark 1. There are only two variables a_1, and a_0 out of 2^n amplitudes. The normalization condition in Equation (16) further reduces the number of variables to 1. $\beta > 1$ is a free parameter being introduced, so all amplitudes can be expressed in terms of β.

Remark 2. To make the geometric series in Equation (10) clearer, we will list the first few terms:

$$\begin{aligned} a_1(k=0) &= \beta^0 a_0(k=0) \\ a_1(k=1) &= \beta^1 a_0(k=1) \\ a_1(k=2) &= \beta^2 a_0(k=2) \end{aligned} \tag{13}$$

Sample values for β are: 2, $2^{1/2}$, $2^{1/n}$, …. Take $\beta = 2$, for example, then:

$$\begin{aligned} a_1(k=0) &= a_0(k=0) \\ a_1(k=1) &= 2a_0(k=1) \\ a_1(k=2) &= 4a_0(k=2) \end{aligned} \tag{14}$$

Proof. From the assumption in Equations (9) and (10):

$$a_1(k) = \beta^k a_0(k) \tag{15}$$

The normalization condition is:

$$\langle \psi_k | \psi_k \rangle = \sum_{x=0}^{N-1} a^*(x,k) a(x,k) = 1 \tag{16}$$

The N amplitudes are divided into two groups: the marked state and all other states. Except for the marked state, the amplitudes for all other states are the same. By applying Equation (8), *i.e.*, there are only two different amplitudes, Equation (16) becomes:

$$\left(a_1(k)\right)^2 + \left(a_0(k)\right)^2 \left(2^n - 1\right) = 1 \tag{17}$$

Using Equation (15):

$$\left(\beta^k a_0(k)\right)^2 + \left(a_0(k)\right)^2 \left(2^n - 1\right) = 1 \tag{18}$$

Solving for $a_0(k)$, we have:

$$a_0(k) = \frac{1}{\sqrt{2^n + \beta^{2k} - 1}}$$

From Equations (12) and (15):

$$a_1(k) = \frac{\beta^k}{\sqrt{2^n + \beta^{2k} - 1}}$$

Theorem 2. Let Amplitude Amplifications be repeated for a certain number of iterations based on Equations (8), (11), and (12), then Grover's algorithm will stop in polynomial time in n, where $n = \log N$.

Proof. The threshold for our algorithm to stop is the same as Grover's algorithm, *i.e.* the amplitude of the marked state $\geq 1/\sqrt{2}$ and the probability $\geq 1/2$. The threshold for our algorithm to stop at step $k = K$ is:

$$\left(a_1(K)\right)^2 = \frac{1}{2} \tag{19}$$

where K is the number of iterations and at $k = K$, the probability of the marked state is 1/2. From amplitude Equation (11) and threshold Equation (19),

$$\left(\frac{\beta^K}{\sqrt{2^n + \beta^{2K} - 1}}\right)^2 = \frac{1}{2} \tag{20}$$

Solve it for K,

$$K = \frac{n}{2\log\beta} = \frac{1}{2\log\beta}\log N \tag{21}$$

where the log base is 2. This is a polynomial time in n and logarithmic time in N. Note that β is a free parameter that is a constant.

Corollary 2.1. If $\beta = 2$,

$$K = \frac{n}{2} = \mathrm{O}(n) = \mathrm{O}(\log N) \tag{22}$$

Corollary 2.2. If $\beta = 2^{1/2}$,

$$K = n = \mathrm{O}(n) = \mathrm{O}(\log N) \tag{23}$$

Corollary 2.3. If $\beta = 2^{1/n}$,

$$K = \frac{n^2}{2} = \mathrm{O}(n^2) = \mathrm{O}(\log^2 N) \tag{24}$$

Proof. By simply inserting the β values into Equation (21), we will get Equations (22), (23), and (24), all of which are polynomial time in n.

Smaller β values will produce a worse time complexity; however, smaller β values will make smaller changes in each step k, which should make the hardware implementation easier. Note that K is the number of iterations, which is the Time Complexity. Theorem 2 and Corollary 2.1, 2.2, 2.3 support the claim regarding O(log$N$) time complexity of the first proposed algorithm.

5. Visualizations

Let us illustrate the basic ideas. We will start with 3 dimensions and then we will expand to N-dimensions. In Figure 1, the three directions represent three amplitudes. Assuming the up-direction state is marked by the Oracle, and the starting superposition vector is Point A, at $k = 0$, which is given by Equation (5),

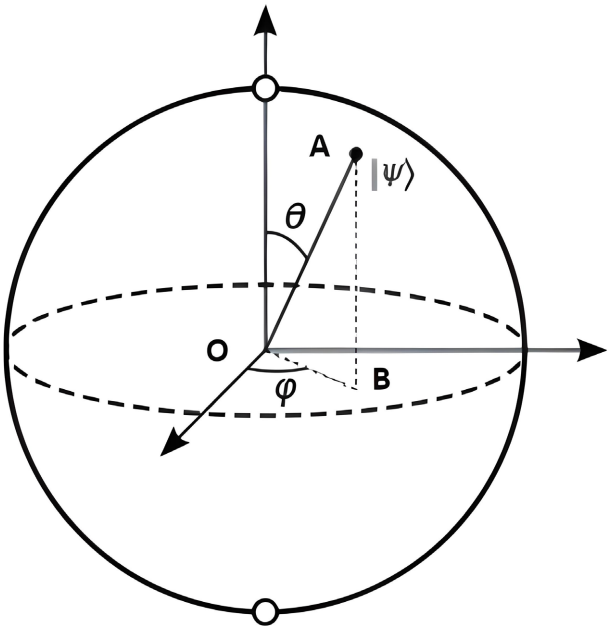

Figure 1. Visualization of the Amplitude Amplification in 3-dimensions. Picture taken from Wikipedia with permission.

then all of the three amplitudes are equal at $k = 0$.

Point B has equal distance to all states not marked by the Oracle. In 3-dimensions, Point B has equal distance to the two unmarked states, so φ is 45 degrees. Point O is the origin. In the OAB plan, at each step in Equation (10), $k = 1, 2, 3, \ldots$, the θ angle within the OAB-plan is decreased, and the amplitude of the marked state is increased because the amplitude of the marked state is $\cos(\theta)$. The amplitude of the other two states is decreased equally, because two amplitudes of the unmarked states are: $|\mathrm{OA}|\sin(\theta)\cos(\varphi)$. Figure 2 shows the rotation of the θ angle.

When we go beyond 3-dimensions, there is no longer a visual image, but the basic idea is exactly the same. Point A is the starting superposition vector. Point B has an equal distance to all states not marked by Oracle. In the OAB plan, at each step in Equation (10), $k = 1, 2, 3, \ldots$, the θ angle within the OAB-plan is decreased, and the amplitude of the marked state is increased because the amplitude of the marked state is $\cos\theta$. The amplitude of all other states is decreased equally. There are $2^n - 1$ unmarked amplitudes, and together, they share $|\mathrm{OA}|\sin(\theta)$, which decreases in each iteration.

Example 1. Let $n = 10$, $\beta = 2$, the Amplitude Amplification is given in Table 1. The iteration will stop at $n/2$ for $\beta = 2$. One extra iteration is done for comparison.

Theorem 3: The rotation matrix, defined by Equations (11) and (12), is unitary.

We will not formally prove this theorem. The rotation is confined within a unit sphere and within the OAB plane in Figure 1. It is unitary because all rotation matrices within a unit sphere are unitary. This is a requirement from Quantum Mechanics where all physical measurable operators must be unitary.

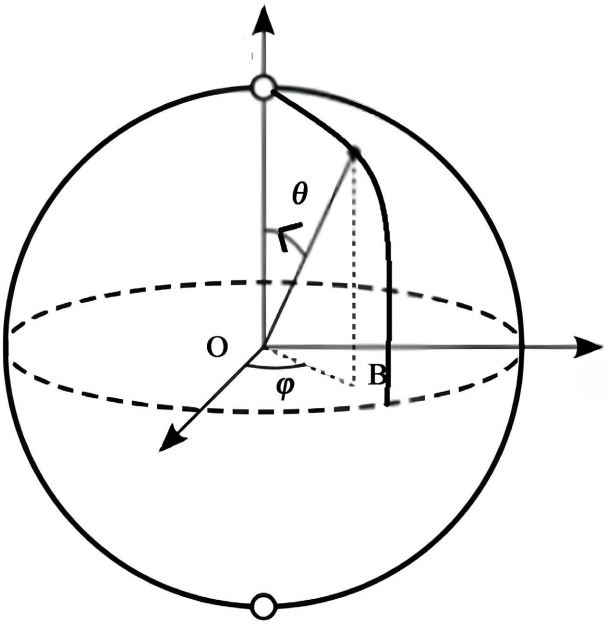

Figure 2. Rotation transformation given by Equations (11) and (12). In each iteration, θ angle within the OAB-plan is decreased, and the amplitude of the marked state is increased. Picture taken from Wikipedia with permission.

Table 1. Amplitude Amplification for $n = 10$ and $\beta = 2$. Column 1, k, represents the iteration number. Column 2, a_1, is the Amplitude Amplification of the marked state. Column 3, θ, shows the rotation in degrees in the OAB-plan. Column 4, a_0, shows that amplitudes of incorrect states are decreased.

k	a_1	θ	a_0
0	0.031	88.209	0.031
1	0.062	86.421	0.031
2	0.124	82.871	0.031
3	0.242	75.957	0.030
4	0.447	63.423	0.027
5	0.707	44.986	0.022
6	0.894	26.553	0.013

6. Amplitude Transfer

In this section, we will introduce a new concept, Amplitude Transfer where the marked state is transferred to a new set of qubits such that the new qubit state is an eigenstate of measurable variables. Assuming a marked state, called the oracle-vector, is:

$$o = \{o_{n-1}, \cdots, o_1, o_0\} \tag{25}$$

i.e. some condition $f(o) = 0$ is satisfied. As we have discussed before, at this point, the probability is $P(o) = 1/N$. Amplitude Amplification is required to change from $P(o) = 1/N$ to $P(o) = 1/2$.

Amplitude Transfer will measure a different set of n qubits, the measurement-vector. Without confusion, we will use the same notation for the qubits in the measurement-vector:

$$x = \{x_{n-1}, \cdots, x_1, x_0\}. \tag{26}$$

There are $2n$ qubits: n qubits in the oracle-vector, which come out of the Oracle black-box, and n-qubits in the measurement-vector. The oracle-vector in Equation (25) is the control qubits. The measurement-vector in Equation (26) is the measurement qubits.

The idea is to transfer the marked state in Equation (25) to the measurement-vector in Equation (26) with high amplitudes. Now we will pair the two groups of qubits so that the control qubits can be applied to some gate circuits:

$$ox = \{(o_{n-1}, x_{n-1}), \cdots, (o_1, x_1), (o_0, x_0)\}. \tag{27}$$

We will initialize the measurement-vector exactly the same way as the Oracle qubits:

$$|\psi'\rangle = |x_{n-1}\rangle \otimes \cdots \otimes |x_1\rangle \otimes |x_0\rangle \tag{28}$$

where

$$|x_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle = H|0\rangle \tag{29}$$

$$|x_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle = H|0\rangle \tag{30}$$

$$\cdots$$

Equation (28) is our starting measurement-vector, *i.e.* we will omit the following Equation in our design in **Figure 4**:

$$|+\rangle = H|0\rangle \tag{31}$$

where H is the Hadamard gate.

Although the measurement-vector in Equation (26) and the oracle-vector in Equation (25) are initialized in the same way, they actually have nothing in common. The oracle-vector is measured in Pauli Z direction. We will measure the measurement-vector in Pauli X direction. Equations (29) and (30) are eigenvectors of Pauli X matrix.

The qubit in the oracle-vector and qubit in the measurement-vector in the Amplitude Transfer is related as follows:

$$|0\rangle \Rightarrow |-\rangle,\ |1\rangle \Rightarrow |+\rangle$$

Example 2.

$$|000\rangle \Rightarrow |---\rangle,\ |001\rangle \Rightarrow |--+\rangle$$

$$|010\rangle \Rightarrow |-+-\rangle, \cdots$$

$$|0000\rangle \Rightarrow |----\rangle,\ |0001\rangle \Rightarrow |---+\rangle$$

$$|0010\rangle \Rightarrow |--+-\rangle, \cdots$$

Equations (28), (29), (30), …, can be rewritten:

$$|\psi'\rangle = |+\rangle \otimes \cdots \otimes |+\rangle \otimes |+\rangle \tag{32}$$

Note that in Equation (32), $|1\rangle \Rightarrow |+\rangle$ is already done; therefore, only the basis state $|0\rangle$ in the oracle-vector will be required to rotate from $|+\rangle$ to $|-\rangle$ in the measurement-vector. There are numerous ways to complete this rotation.

The reason why Amplitude Transfer works is obvious: the measurement-vector is eigenvector or almost an eigenvector along Pauli X direction. When the measurement-vector is measured, the correct solution will be obtained with a probability of 100% or almost 100%.

Before we present the Amplitude Transfer in **Figure 4**, we will list the Gate equation required for our design.

7. The Gate Equations Used

We will first list all the Gate equations used in this paper, which are all well-known [1] [2] [3] [4] [5]. We will explain how they will be used. We will need X gate, Phase shift gates, Controlled Phase shift gates, and Hadamard gate.

The Pauli gate X is:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{33}$$

Note that:

$$X|+\rangle = |+\rangle, \quad X|-\rangle = -|-\rangle \tag{34}$$

The above equation indicates that $|+\rangle$ is an eigenvector of X; in other words, $|+\rangle$ is measurable along Pauli X direction with a probability of 100% for the eigenvalue of 1. Similarly, $|-\rangle$ is measurable along Pauli X direction with a probability of 100% for the eigenvalue of −1.

The Pauli X gate is also a "Not" gate along Pauli Z direction:

$$X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle \tag{35}$$

Equation (35) will be used for the flipping of control qubits.

Phase shift gates are:

$$R(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{i\varphi} \end{bmatrix} \tag{36}$$

where φ is the phase shift. The Phase shift is a family of single-qubit gates that map the basis states:

$$R(\varphi)|0\rangle = |0\rangle, \quad R(\varphi)|1\rangle = \mathrm{e}^{i\varphi}|1\rangle \tag{37}$$

Also,

$$R(\varphi)|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + \mathrm{e}^{i\varphi}|1\rangle\right) \tag{38}$$

The Hadamard gate (H gate) has:

$$H|+\rangle = |0\rangle, \quad H|-\rangle = |1\rangle \tag{39}$$

$$XH|+\rangle = |1\rangle, \quad XH|-\rangle = |0\rangle \tag{40}$$

Throughout this paper, for simplicity, the measurable variables are the Pauli X matrix, *i.e.*, the measurement is along the X-direction. This measurement direction can be changed from X to Z direction by Equations (39) and (40).

The Controlled Phase shift gates are:

$$CR(\varphi) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes R(\varphi) \tag{41}$$

So

$$CR(\varphi)|0\rangle|x\rangle = |0\rangle|x\rangle \tag{42}$$

$$CR(\varphi)|1\rangle|x\rangle = |1\rangle \otimes R(\varphi)|x\rangle \tag{43}$$

Equations (41)-(43) are the main Equations used in the Amplitude Transfer. Figure 3 shows the Controlled Phase shift gates.

8. T = O(log$N$) Amplitude Transfer

In this section, we will implement Amplitude Transfer. We will use a simple example to show the design in Figure 4, and then, generalize the design. Given an unsorted list,

$$M = \{m_0, m_1, \cdots, m_{L-1}\} \tag{44}$$

The unstructured search problem is to find a target in the list. A simple example is finding a target, 5, from arbitrary list, M = {3, 4, 6, 0, 1, 2, 7, 5}. The Oracle will mark target, $|5\rangle = |101\rangle$, which is given in this example.

The Amplitude Transfer will transfer the oracle-vector, $|101\rangle$, to measurement-vector, $|+-+\rangle$.

Again, the measurement direction is Pauli X. Unlike Pauli Z direction, where the base states are $|0\rangle$ and $|1\rangle$; in Pauli Z direction, the base states are $|-\rangle$ and $|+\rangle$. To produce the O(log$N$) algorithm, the number of rotations must be O(n), or simply n. There are numerous ways to achieve this. For example, in Equation (38), let the final angle be $\varphi' = \pi$, then the rotation from $|+\rangle$ to $|-\rangle$ is done:

$$R(\varphi)|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = |-\rangle$$

Let the final angle be $\varphi' = \pi$ and let the number of rotations be n; then each

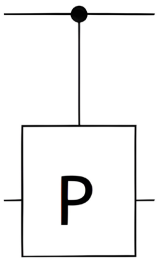

Figure 3. Controlled Phase shift gates.

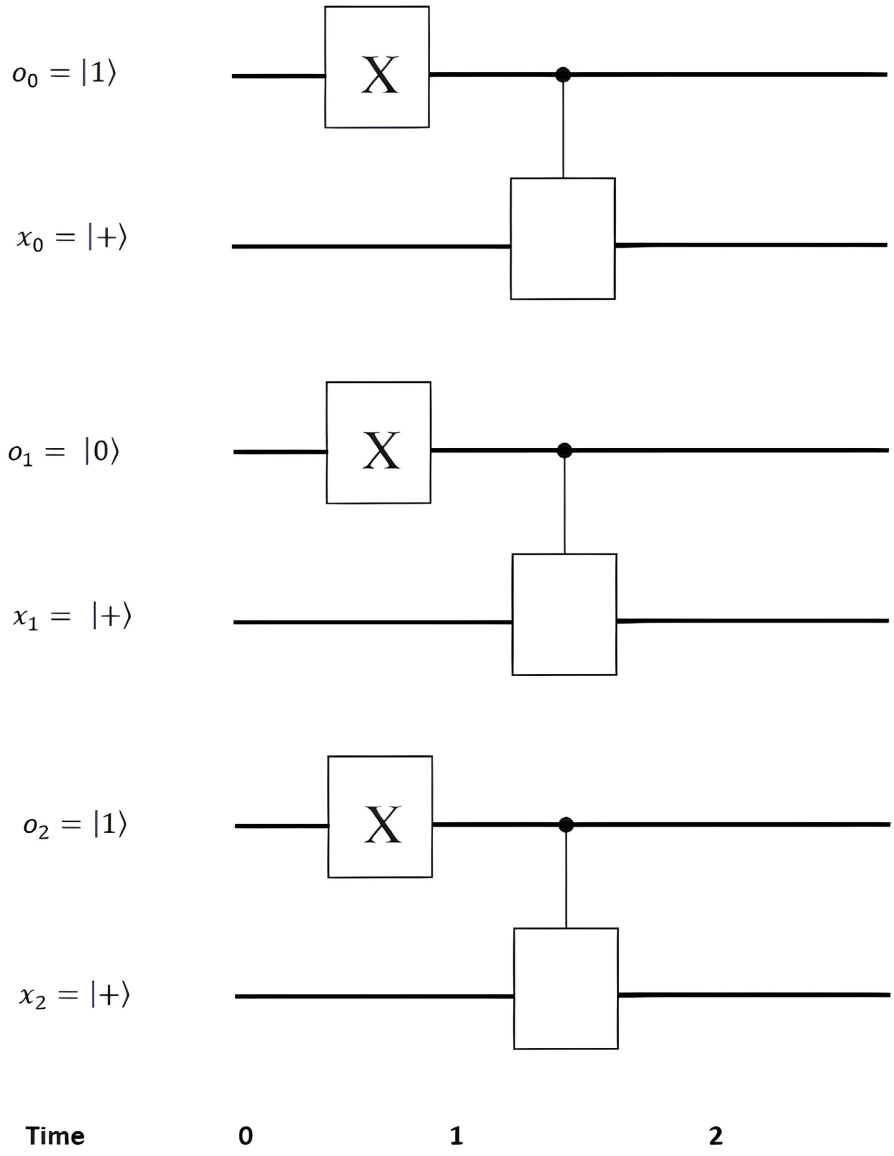

Figure 4. An example of O(log$N$) Circuit Design. The empty gates represent that there are numerous ways to rotate from $|+\rangle$ to $|-\rangle$. An example is 3-rotations: *PPP*, where *P* is given in Equation (36) with $\varphi = \pi/3$. Note that all the qubits proceed in parallel.

rotation is $\pi/n = \pi/3$. This means that the rotation gates are *PPP*, where *P* is given by Equation (36) with value, $\varphi = \pi/3$.

At Step 0 in Figure 4, the oracle-vector is:

$$o = \{o_2 = |1\rangle, o_1 = 0|0\rangle, o_0 = |1\rangle\}.$$

The initial measurement-vector, based on Equations (28)-(30) is:

$$x = \{x_2 = |+\rangle, x_1 = |+\rangle, x_0 = |+\rangle\}.$$

Only the middle qubit needs attention in this example.

At Step 1 (after *X* gate in Figure 3), using Equation (35), the oracle-vector is:

$$o = \{o_2 = |0\rangle, o_1 = |1\rangle, o_0 = |0\rangle\}.$$

Note that the X gates flipped the control qubits from 101 to 010. This step selects the control qubit so that only the middle qubit is chosen. This will prepare the next step, which will change the measurement-vector from +++ to +−+, where 7 (+++) is the initial state and 5 (+−+) is the final state. Again, Pauli X is the measurement direction.

At Step 2, using Equations (41), (42) and (43), the PPP is only applied to the middle qubit. From Equation (38), we have:

$$|-\rangle = PPP|+\rangle.$$

The measurement-vector is:

$$x = \{x_2 = |+\rangle, x_1 = |-\rangle, x_0 = |+\rangle\}.$$

Note that at this point, PPP gates rotated x_1 qubit from $x_1 = |+\rangle$ to $x_1 = |-\rangle$. All three qubits are the eigenvectors of Pauli X matrix; so, if we take a measurement at this point, the only outcome is 101.

At Step 3, take a measurement, the probability $P(|101\rangle)$ is 1, and the target, 5, is identified.

Let's us have another example, assuming the problem is finding a target, 2, from $M = \{3, 4, 6, 0, 1, 2, 7, 5\}$.

At Step 0,

$$o = \{o_2 = |0\rangle, o_1 = 0|1\rangle, o_0 = |0\rangle\}$$

$$x = \{x_2 = |+\rangle, x_1 = |+\rangle, x_0 = |+\rangle\}$$

At Step 1,

$$o = \{o_2 = |1\rangle, o_1 = |0\rangle, o_0 = |1\rangle\}$$

At Step 2,

$$x = \{x_2 = |-\rangle, x_1 = |+\rangle, x_0 = |-\rangle\}$$

At Step 3, take a measurement, the probability $P(|010\rangle)$ is 1, and the target, 2, is identified.

Comment 1: Although the measurement-vector is measured with a high probability to obtain the correct solution, if the oracle-vector is measured, it has a very low probability to obtain the correct solution. In fact, the oracle-vector has a distribution of Equation (5). Oracle and Amplitude Amplification fulfill a different role.

Comment 2: Equation (40) will allow changing the measurement direction from X to Z by inserting XH gates at the end of **Figure 4** for x-qubits, so the measurement direction is not important theoretically, although it could be very important in the implementation of the circuit.

Generalization from **Figure 4**, with 3 qubits to n qubits, is trivial: simply repeat the Controlled Phase shift gate n times and replace the empty box in **Figure 4** by P^{n} and set the rotation angle for each P to be:

$$\varphi = \frac{\pi}{n} \tag{45}$$

The input for Figure 4 is Equation (27), which consists of the oracle-vector in Equation (25) and the measurement-vector in Equation (26). The oracle-vector is the output of the Oracle black-box. The output from Figure 4 is a measurement-vector where the correct solution is, or almost is, an eigenvector.

The Time Complexity analysis is straightforward. Note that in Figure 4, all the qubits proceed in parallel so we can simply examine one qubit. Because of n steps in phase shift gates: P^n, the time complexity is $T = O(n) = O(\log N)$.

Again, in Figure 4, there are numerous ways to rotate from $|+\rangle$ to $|-\rangle$.

9. Modified Unstructured Search

The paper introduces two new algorithms for Amplitude Amplification in Grover's algorithm, reducing the time complexity from $O(\sqrt{N})$ to $O(\log N)$. Through the novel approaches that include geometric series and Amplitude Transfer, this paper aims to enhance the efficiency of quantum computing in problems such as unstructured search problems.

The two new search algorithms proposed in this paper are:

- Initialization;
- Oracle;
- $O(\log N)$ Amplitude Amplification/Transfer;
- Measurement.

Let the above algorithms be applied to the unsorted search:

- The Initialization given in Equation (5) is O(1);
- The Oracle is O(1) because it simply passes the target through the Oracle black box;
- The Amplitude Amplification is $O(\log N)$, which is the purpose of this paper; and
- The measurement is O(1).

So the unsorted-list search, based on Amplitude Amplification alone, has a time complexity of $O(\log N)$.

10. Discussions

In Computer Science and Computational Complexity Theory [9] [10] [11] [12], P and NP are classes of decision problems. P = Polynomial Time and NP = Nondeterministic Polynomial Time. P = NP is one of the most famous unsolved problems in Computer Science and Mathematics. There is no proof that P = NP, and the prevailing belief among most experts is well-known: P < NP.

Quantum Computing leverages the principles of Superposition and Entanglement. An O(1) step in a quantum computer can be simulated by a classical computer in $O(2^n)$ steps, where n is the number of qubits. This is because of Equation (5): simply going through Equation (5), the classical time complicity is $O(N) = O(2^n)$. Reversely, in some situations, $O(2^n)$ steps in a classical computer can be simulated in a quantum computer in O(1) steps. This provides a potential for an $O(2^n)$ speed up in time complexity. This is the potential power of quan-

tum parallelism.

If an $O(2^n)$ speed could be applied on the NP computation class, then the time complexity of an NP problem would run in polynomial time. This would have significant implications, as it would imply that many computationally difficult problems [9] [10] [11] [12], like the traveling salesman problem, the Boolean satisfiability problem, etc. could be solved in a quantum computer with Polynomial times.

If we apply the two algorithms in Section 9 to an NP problem, we have reduced the time complexity of NP problems to the time complexity of the Oracle black box, because in the algorithm:

- The Initialization is O(1);
- The Amplitude Amplification is O(log*N*), which is the purpose of this paper;
- The measurement is O(1).

Oracle is a big topic, and we will discuss the Oracle designs in our other papers [13].

In terms of implementation of the algorithms, both Grover's Amplitude Amplification and our first algorithm represent a unitary rotation in a unit sphere. Section 5 shows a visual image for these rotations. The practical implementations of both algorithms are equally challenging. Our second algorithm, on the other hand, uses only well-known single qubit gates, so it has the advantages of scalability, robustness, and applicability to many different problems in quantum computing. The current limitations in quantum computing hardware are in the order of 100 qubits, half of which can be used directly for computation (our algorithm requires 2n qubits). The latest system, run by Google, has a total of 70 operational qubits [14]. This means that the applicability of the proposed algorithms for solving various problems are quite practical, especially the Amplitude Transfer algorithm.

11. Conclusions

In this paper, we have proposed an O(log*N*) Algorithm for Amplitude Amplification and O(log*N)* Algorithm for Amplitude Transfer in Grover's Algorithm, which have significantly reduced the Amplitude Amplification time.

The difference between Grover's algorithm and our first algorithm is that the Amplitude Amplification ratio in Grover's algorithm is an arithmetic series and ours, is a geometric one. Because our Amplitude Amplification ratios converge much faster, the time complexity is improved significantly.

In our second algorithm, we introduced a new concept, Amplitude Transfer where the marked state is transferred to a new set of qubits such that the new qubit state is an eigenstate of measurable variables. When the new qubit quantum state is measured, with high probability, the correct solution will be obtained.

Acknowledgements

I would like to thank Gina Porter for proof-reading this paper.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.

[2] Rieffel, E. and Polak, W. (2011) Quantum Computing: A Gentle Introduction. The MIT Press, Cambridge.

[3] Johnston, E.R., Harrigan, N. and Gimeno-Segovia, M. (2019) Programming Quantum Computers: Essential Algorithms and Code Samples. O'Reilly Media, Sebastopol.

[4] Hidary, J.D. (2019) Quantum Computing: An Applied Approach. Springer, Cham. https://doi.org/10.1007/978-3-030-23922-0

[5] Aaronson, S. (2013) Quantum Computing Since Democritus. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511979309

[6] Grover, L.K. (1996) A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (*STOC*'96), 212-219. https://doi.org/10.1145/237814.237866

[7] Berti, A. (2022) Behind Oracles: Grover's Algorithm. https://towardsdatascience.com/behind-oracles-grovers-algorithm-amplitude-amplification-46b928b46f1e

[8] Shirgure, S. (2024) Solving the Subset Sum Problem on a Quantum Computer. https://sumeetshirgure.github.io/assets/pdfs/presentations/ee_520.pdf

[9] Arora, S. and Barak, B. (2009) Computational Complexity: A Modern Approach. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511804090

[10] Papadimitriou, C.H. (1994) Computational Complexity. Addison-Wesley, Boston.

[11] Garey, M.R. and Johnson, D.S. (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York.

[12] Sipser, M. (2012) Introduction to the Theory of Computation. Cengage Learning, Boston.

[13] Liu, Y. (2024) Time Complexity of the Oracle Phase in Grover's Algorithm. *American Journal of Computational Mathematics*, **14**, 1-10. https://doi.org/10.4236/ajcm.2024.141001

[14] Nield, D. (2024) Google Quantum Computer Is '47 Years' Faster Than #1 Supercomputer. Science Alert. https://www.sciencealert.com/google-quantum-computer-is-47-years-faster-than-1-supercomputer

Appendix: Geometric Series vs Arithmetic Series Comparison

The underlying methodology of our first algorithm is that our geometric series stops much faster than Grover's arithmetic series. From Equations (9) and (10), our Amplitude Amplification ratio is a geometric series:

$$r = \beta^0, \beta^1, \beta^2, \cdots, \beta^K, \ \ \beta > 1 \tag{46}$$

In this Appendix, we will demonstrate that the Amplitude Amplification ratio in Grover's algorithm is an arithmetic series by iterations.

In Grover's algorithm, Amplitude Amplification is achieved through repeated applications of two main steps:

- Oracle Operation: The Oracle marks the target state by inverting its amplitude, which distinguishes the target state from the rest of the states.
- Inversion with Respect to the Mean: After applying the Oracle operation, the algorithm applies a transformation known as inversion with respect to the mean. This step reflects the amplitude distribution of all states with respect to the average amplitude, effectively amplifying the amplitude of the target state and reducing the amplitudes of other states.

Again, there are only two different amplitudes because of Equation (8). To distinguish Grover's algorithm from our first algorithm, we will use a different notation to define the Amplitude Amplification ratio between the marked state and other states:

$$\alpha(k) = \frac{a_1(k)}{a_0(k)} \tag{47}$$

The detailed operation in a single step is:

- Inverting the amplitude of the marked state;
- Computing the mean, and shifting the amplitude coordinate system;
- Reflecting all amplitudes;
- Shifting the amplitude coordinate system back.

Below, let α be the ratio at step k and we will follow this process for one iteration and compute the amplification for one iteration.

Inverting the amplitude of the marked state:

$$a_0' = a_0, \ \ a_1' = -a_1 \tag{48}$$

Computing the mean:

$$avg = \frac{1}{2^n}\left(a_0' + a_1' + a_2' + a_3' + \cdots\right) \tag{49}$$

Using the average in Equation (49), amplitude ratio definition in Equation (47), amplitude assumption in Equation (8), and inverting in Equation (48), we have:

$$avg = a_0'\left(1 - \frac{\alpha + 1}{2^n}\right) \tag{50}$$

Shift the amplitude coordinate system:

$$a_0'' = a_0' - avg = a_0\left(\frac{\alpha+1}{2^n}\right) \tag{51}$$

$$a_1'' = a_1' - avg = a_0\left[-(\alpha+1)+\frac{\alpha+1}{2^n}\right] \tag{52}$$

Reflecting all amplitudes:

$$a_0'' = -a_0\left(\frac{\alpha+1}{2^n}\right) \tag{53}$$

$$a_1'' = a_0\left[(\alpha+1)-\frac{\alpha+1}{2^n}\right] \tag{54}$$

Shift the amplitude coordinate system back:

$$a_0''' = a_0'' + avg = a_0\left(1-\frac{2(\alpha+1)}{2^n}\right) \tag{55}$$

$$a_1''' = a_1'' + avg = a_0\left[(\alpha+2)-\frac{2(\alpha+1)}{2^n}\right] \tag{56}$$

By applying these steps for one iteration, Grover's algorithm has increased the probability amplitude of the target state while decreasing the amplitudes of other states by the following amount:

$$\frac{a_1'''}{a_0'''} = 1+\frac{1+\alpha}{1-\frac{2(\alpha+1)}{2^n}} \tag{57}$$

For large n,

$$\frac{2(\alpha+1)}{2^n} \ll 1$$

So,

$$\frac{a_1'''}{a_0'''} \le 1+\frac{1+\alpha}{1}$$

Here α is the current Amplitude Amplification ratio between the marked state and other states, and the LHS is the ratio after one iteration, $\alpha(k+1)$. The iterative equation for α is:

$$\alpha(k+1) \le 2+\alpha(k) \tag{58}$$

At $k=0$, the amplitudes are given by Equation (5), so

$$\alpha(0) = 1.$$

At $k=1$,

$$\alpha(1) \le 2+\alpha(0) = 3$$

At $k=2$,

$$\alpha(2) \le 2+\alpha(1) = 5$$

…

The Amplitude Amplification ratio between the marked state and other states

is an arithmetic series:

$$1,3,5,7,\cdots,2K+1 \tag{59}$$

This arithmetic series leads to O($\sqrt{N}$) Time Complexity [6].