Journal of Quantum Information Science, 2025, 15(4), 208-215
https://www.scirp.org/journal/jqis
ISSN Online: 2162-576X
ISSN Print: 2162-5751

An Improved Quantum Search Algorithm

Amrita Mitra

Bangalore, India
Email: mitra.amrita2017@gmail.com

How to cite this paper: Mitra, A. (2025) An Improved Quantum Search Algorithm. *Journal of Quantum Information Science*, **15**, 208-215.
https://doi.org/10.4236/jqis.2025.154012

Received: November 7, 2025
Accepted: December 16, 2025
Published: December 19, 2025

Abstract

If we have to search an unsorted database or solve an unstructured search problem, we can use a quantum search algorithm [1]. As of now, a quantum search algorithm like Grover's algorithm [2] takes $\Omega\left(\sqrt{N}\right)$ time, where N is the size of the search space. This paper proposes an improved quantum search algorithm that yields the correct result in constant time with a high probability.

Keywords

Quantum Algorithm, Quantum Search, Quantum Computing

1. Introduction

Let's say $f:\{0,1,2,\cdots,N-1\}\to\{0,1\}$ is a function. In the case of an unstructured database, the domain of the function represents the indices of the database. $f(x)$ can be 0 or 1. $f(x)=1$ indicates that x points to one index of the database that contains the element that satisfies the search criteria. In all other cases, $f(x)=0$. Our purpose is to find the index x for which $f(x)=1$.

2. The Proposed Quantum Algorithm

Let's say N is the size of the search space. Let's also assume that $n=\lceil \log_2 N \rceil$. Therefore, the binary representation of an index needs a maximum n qubits.

For our proposed algorithm, we can use the quantum circuit as shown in Figure 1. The first n qubits contain all zeros. And the ancillary qubit contains a 1. A Hadamard gate is applied to the first n qubits. A Hadamard gate is also applied to the ancillary qubit. We get the following state as a result:

$$|\psi_1\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle \frac{\left(|0\rangle - |1\rangle\right)}{\sqrt{2}} \tag{1}$$

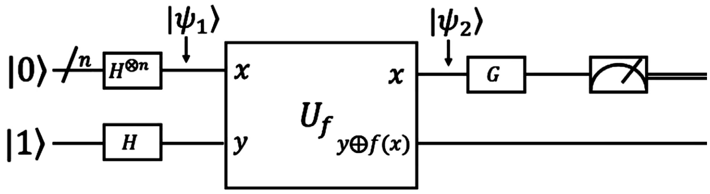

Figure 1. Quantum circuit for the proposed quantum algorithm.

In our proposed algorithm, we use an oracle U_f. If $|x\rangle$ is given as input and $|y\rangle$ is the ancillary qubit, then we get the following state on querying the oracle:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|y\oplus f(x)\rangle \tag{2}$$

$\oplus$ is the addition modulo 2 operator. In our case, the ancillary qubit is 1, and one Hadamard gate is applied to it. Therefore, on querying the oracle U_f, we get the following state [3]:

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\frac{|0\oplus f(0)\rangle-|1\oplus f(1)\rangle}{\sqrt{2}} \\ &= \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{f(x)}|x\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}\end{aligned} \tag{3}$$

For example, if $N=4$, $n=\log_2 N=2$. In that case, the state $|\psi_1\rangle$ is going to be the following:

$$\begin{aligned}|\psi_1\rangle &= \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}} \\ &= \frac{(|00\rangle+|01\rangle+|10\rangle+|11\rangle)(|0\rangle-|1\rangle)}{2\sqrt{2}}\end{aligned} \tag{4}$$

After querying the oracle, we will get the following state:

$$|\psi_2\rangle = \frac{1}{2}\sum_{x=0}^{3}(-1)^{f(x)}|x\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}} \tag{5}$$

Let's consider $f(x)=1$ when $x=0$. And in all other cases, $f(x)=0$. Therefore, the state $|\psi_2\rangle$ is going to be the following in this case:

$$|\psi_2\rangle = \frac{(|00\rangle-|01\rangle-|10\rangle-|11\rangle)}{2}\times\frac{(|1\rangle-|0\rangle)}{\sqrt{2}} \tag{6}$$

Therefore, the first n qubits after querying the oracle are the following:

$$\frac{(|00\rangle-|01\rangle-|10\rangle-|11\rangle)}{2} \tag{7}$$

We can now apply a quantum gate G on this state, so that the output of this gate G is $|00\rangle$. Please note that in our example, $f(x)=1$ when $x=00$. For any other value of x, $f(x)=0$.

$$|\psi_3\rangle = |00\rangle \tag{8}$$

If we measure $|\psi_3\rangle$, we will get the index in the unsorted database that points to the entry that satisfies the search criteria.

3. Implementation of the Quantum Gate G

Let's say M is a unitary matrix that implements the quantum gate G. In this section, we will attempt to determine the unitary matrix M.

Let's say N is the size of the search space. The indices in the unstructured database are $0,1,\cdots,N-1$. It will take $n=\lceil \log_2 N \rceil$ qubits to represent these N indices.

Let the indices in the unstructured database be $x_0,x_1,\cdots,x_{N-1}$. And let the corresponding values of $f(x)$ be $y_0,y_1,\cdots,y_{N-1}$.

For example, in our example, the binary indices in the unstructured database are $00,01,10,11$. The corresponding values of $f(x)$ are the following:

$$f(00)=1$$

$$f(01)=0$$

$$f(10)=0$$

$$f(11)=0$$

This indicates the value pointed to by the binary index 00 satisfies the search criteria, and all other values do not. As a result, the first n qubits after querying the oracle are the following:

$$\frac{\left(|00\rangle-|01\rangle-|10\rangle-|11\rangle\right)}{2} \tag{9}$$

We want the output of the quantum gate G to be $|00\rangle$.

Let's say the column vector X represents the first n qubits after querying the oracle.

$$X=\frac{1}{\sqrt{N}}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} \tag{10}$$

In our example, X represents the following state:

$$\frac{\left(|00\rangle-|01\rangle-|10\rangle-|11\rangle\right)}{2} \tag{11}$$

Therefore, in our example,

$$X=\frac{1}{\sqrt{N}}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}=\frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \tag{12}$$

Let the column vector Y represent the expected output.

$$Y=\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} \tag{13}$$

For example, in our example, the expected output is $|00\rangle$. Therefore, Y is the following column vector.

$$Y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{14}$$

In other words, the unitary matrix M is applied to X, and we get the output Y.

$$MX = Y \tag{15}$$

Therefore, if we assume $g:\{-1,0,1\} \to \{0,1\}$ to be a function that takes each element of X as input and gives the corresponding element of Y as output, then the following holds true:

$$g(-1) = 0$$

$$g(0) = -1$$

$$g(1) = 1$$

Therefore,

$$\begin{aligned} &\text{When } x = -1,\ g(g(x)) = g(g(-1)) = g(0) = -1 \\ &\text{When } x = 0,\ g(g(x)) = g(g(0)) = g(-1) = 0 \\ &\text{When } x = 1,\ g(g(x)) = g(g(1)) = g(1) = 1 \end{aligned} \tag{16}$$

Therefore, g is a function for which $g(g(x)) = x$. Therefore, g is an involution function. And the corresponding matrix M is an involutory matrix.

4. Properties of Matrix *M*

Property 1: M is a unitary matrix.

As M implements a quantum gate, M must be a unitary matrix [4]. For a unitary matrix U, the following holds true:

$$U^* U = UU^* = I \tag{17}$$

In our case, M is a real matrix. Therefore,

$$M^{\mathrm{T}} M = MM^{\mathrm{T}} = I \tag{18}$$

Property 2: M is an involutory matrix.

As M is an involutory matrix, we can write the following:

$$M = M^{-1} \tag{19}$$

Therefore,

$$M^2 = MM^{-1} = I \tag{20}$$

Property 3: M is an orthogonal matrix.

As M is a unitary matrix and an involutory matrix, we can write the following:

$$M^{\mathrm{T}} M = M^{-1} M = I = MM^{-1} = MM^{\mathrm{T}} \tag{21}$$

Therefore,

$$M^{\mathrm{T}} = M^{-1} \tag{22}$$

Therefore, M is an orthogonal matrix.

Property 4: M is a symmetric matrix.

As M is an orthogonal matrix, we can write the following:

$$M^{\mathrm{T}}M = I \tag{23}$$

$$M^{\mathrm{T}} = M^{-1} \tag{24}$$

As M is an involutory matrix, we can write the following:

$$M = M^{-1} \tag{25}$$

Therefore,

$$M^{\mathrm{T}} = M \tag{26}$$

Therefore, M is a symmetric matrix.

Based on the above properties, we will look for the value of the matrix M that is as close to X^{-1} as possible. As the value of M is not equal to X^{-1} for all values of N, the value of M is going to introduce an error term. We will discuss the error term in Section 6.

5. Finding the Value of *M*

Ideally, we are looking for a matrix M for which the following condition holds true:

$$X = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & -1 & \cdots & -1 \\ -1 & 1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & 1 \end{bmatrix} \tag{27}$$

$$Y = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = I \tag{28}$$

$$MX = I \tag{29}$$

Therefore, M should be equal to X^{-1}. However, X^{-1} may not be a unitary matrix for all values of N. Therefore, we will try to find a unitary matrix M such that MX is as close to I as possible.

Let's say,

$$M = X^{-1} + E \tag{30}$$

$$\therefore MX = \left(X^{-1} + E\right)X = I + EX \tag{31}$$

This EX is the error term.

Therefore, we will look for an $N \times N$ matrix M for which the following conditions hold true:

1) $MM^{\mathrm{T}} = M^{\mathrm{T}}M = I$.

2) $MM^{-1} = M^{-1}M = I$.

3) $M^{\mathrm{T}} = M^{-1}$.

4) $\det(M) = \pm 1$ as M is involutory.

5) $M^2 = I$.

6) The error term EX is minimum.

We observe the following:

$$X = \frac{1}{\sqrt{N}}(2I - J) \tag{32}$$

$$J = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{N\times N} \tag{33}$$

Using the Sherman-Morrison formula [5], we get the following:

$$X^{-1} = \frac{\sqrt{N}}{2} I - \frac{\sqrt{N}}{2N-4} J \tag{34}$$

Let's say M is the following matrix:

$$M = k_1 I - k_2 J \tag{35}$$

Using the Sherman-Morrison formula, we get the following:

$$M^{-1} = \frac{1}{k_1} I - \frac{k_2}{k_1^2 (Nk_1k_2 - 1)} J \tag{36}$$

As M is a symmetric, unitary, and involutory matrix, we get the following:

$$M^{-1} = M$$

$$\therefore \frac{1}{k_1} I - \frac{k_2}{k_1^2 (Nk_1k_2 - 1)} J = k_1 I - k_2 J \tag{37}$$

By comparing both sides, we get the following:

$$k_1 = 1$$

$$k_2 = \frac{k_2}{k_1^2 (Nk_1k_2 - 1)}$$

$$\therefore k_2 = \frac{2}{N}$$

$$\therefore M = k_1 I - k_2 J = I - \frac{2}{N} J \tag{38}$$

Here, I is the identity matrix, and J is a matrix with all entries equal to 1. Therefore, the diagonal entries of the matrix M are $\left(1 - \frac{2}{N}\right)$ and the non-diagonal entries of M are $-\frac{2}{N}$. Given the value of N, these values can be computed in $O(1)$ time. The value of M does not depend on a specific input.

6. The Error Term

As we discussed, the error term is EX. We observe the following:

$$X = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & -1 & \cdots & -1 \\ -1 & 1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & 1 \end{bmatrix} = \frac{1}{\sqrt{N}}(2I - J)$$

$$J = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{N\times N}$$

$$M = X^{-1} + E$$

$$\begin{aligned} \therefore E &= \left(M - X^{-1}\right) \\ &= I - \frac{2}{N}J - \left(\frac{\sqrt{N}}{2}I - \frac{\sqrt{N}}{2N-4}J\right) \\ &= \left(1 - \frac{\sqrt{N}}{2}\right)I - \left(\frac{2}{N} - \frac{\sqrt{N}}{2N-4}\right)J \end{aligned}$$

$$\therefore EX = \frac{1}{\sqrt{N}}\left(\left(1 - \frac{\sqrt{N}}{2}\right)I - \left(\frac{2}{N} - \frac{\sqrt{N}}{2N-4}\right)J\right)(2I - J) \tag{39}$$

7. Results

Let's say N is the search space. n is the required number of qubits.

$$\therefore N = 2^n$$

If we calculate the values of M, X, X^{-1}, E, and EX for different values of n, and take the determinant of the error matrix EX, we get the following Table 1:

Table 1. Determinant of the error term.

n	$N = 2^n$	$\det(EX)$
3	8	−0.00020734564555665073
4	16	−7.629394531249957e−05
5	32	−5.758564428031929e−06
6	64	−9.081620849843095e−08
7	128	−1.89081434676154e−10
8	256	−2.4238765388563638e−14
9	512	−6.252713627319217e−20
10	1024	−6.562978374539346e−28
11	2048	−2.8990010884419623e−39

Please note that the output matrix Y is the following:

$$Y = MX = \left(I - \frac{2}{N}J\right) \times \frac{1}{\sqrt{N}}(2I - J) = \frac{1}{\sqrt{N}}\left(2I + \frac{N-4}{N}J\right) \tag{40}$$

In other words, the diagonal entries of the output matrix Y will be the following:

$$\frac{1}{\sqrt{N}}\left(2+\frac{N-4}{N}\right)=\frac{1}{\sqrt{N}}\times\frac{3N-4}{N}=\frac{3N-4}{N\sqrt{N}} \tag{41}$$

And, the non-diagonal entries will be the following:

$$\frac{N-4}{N\sqrt{N}} \tag{42}$$

Therefore, once we find the required matrix M, the first n qubits of the following state are applied to the corresponding quantum gate:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{f(x)}|x\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}} \tag{43}$$

As output, we will obtain the index from which we can retrieve the positions of the elements in the subset that satisfy the search criteria, along with the error as stated above.

8. Conclusions

Until now, the time complexity of a quantum search algorithm is $\Omega(\sqrt{N})$ where N is the size of the search space. This paper proposes a quantum algorithm that reduces this time complexity. The proposed quantum algorithm requires n qubits as input, where $n=\lceil\log_2 N\rceil$, and one ancillary qubit. After measuring the n-qubit output of the quantum gate G implemented using the unitary matrix M, we get the index in the unsorted database that satisfies the search criteria with a significantly high probability.

Though the unitary matrix M corresponding to the quantum gate G is an $N\times N$ matrix, there are only two different values for elements of M. All the diagonal elements of M have the same value. All non-diagonal elements of M have the same value. Therefore, the matrix M can be constructed in $O(1)$ time.

Conflicts of Interest

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

References

[1] Pittenger, A.O. (2012) An Introduction to Quantum Computing Algorithms. Vol. 19, Springer Science & Business Media.

[2] Grover, L.K. (1996) A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, New York, 1 July 1996, 212-219. https://doi.org/10.1145/237814.237866

[3] Nielsen, M.A. and Isaac L.C. (2010) Quantum Computation and Quantum Information. Cambridge University Press.

[4] Horn, R.A. and Johnson, C.R. (2012) Matrix Analysis. Cambridge University Press.

[5] Wikipedia Contributors (2025) Sherman-Morrison Formula. Wikipedia, Wikimedia Foundation.