

---

# A QUANTUM IMPROVEMENT ON DIJKSTRA'S ALGORITHM FOR COMPUTER NETWORK ROUTING

Ahmed T. Sadik

Rana F. Ghani



Department of Computer Science-University of Technology

---

## ARTICLE INFO

Received: 1 / 4 /2008  
Accepted: 24 / 4 /2008  
Available online: 30/4/2008  
DOI: [10.37652/juaps.2008.15462](https://doi.org/10.37652/juaps.2008.15462)

### Keywords:

Quantum Computing,  
Routing Algorithms,  
Computer Network.

## ABSTRACT

The aim of this paper is to improve the Dijkstra algorithm which is widely used in the internet routing. Quantum computing approach is used to improve the work of Dijkstra algorithm for network routing by exploiting the massive parallelism existing in the quantum environment and to deal with the demands of continuous growing of the internet. This algorithm is compared according to the number of iterations and time complexity with Dijkstra's algorithm and the result shows that the quantum approach is better in finding the optimal path with better time complexity when it is implemented in quantum computer.

---

## Introduction

The basic technique of the Internet, that binds these thousands of component networks into a single Internet, is addressing and routing. In the address part, each network uses a unique set of addresses drawn from a single global address space. Each connected device has a unique address that it uses to label its network interface. Each portion of information (packet) generated by these devices has a source and destination address. The source address references the local interface address, and, logically, the destination address is the corresponding interface address of the intended recipient. As it is being passed within the network from router to router, the router can identify this intended recipient. But within a network identity is only half of the solution. The network must be able to know location, or, where the packet is to be directed.

The task of associating location with identity, or in other words maintaining routing information within a network is undertaken by routing protocols which include routing algorithms that determine the best path to send a packet from its source to destination [4].

## Quantum Computation

In the 20th century great achievements were made within the field of computation. The researches in the field of quantum mechanics made it possible to create transistors and microchips thereby opening the door for the high speed personal computers. However, even the building of such computers is in its first steps, the scientist have realized that when such a computer is being made possible the operations are still classical (i.e. the implementation techniques of information theory becomes quantum mechanical, while the theory itself remains classical).

The difference between classical computation and quantum computation is that the classical computer system is always is one of defined mutually exclusive

---

\* Corresponding author at: Department of Computer Science-  
University of Technology, Iraq  
.E-mail address:

set of states, while a quantum system can exist in what is known as superposition state. Hence, a quantum computer can process different input in parallel and produces a superposition of outputs [3].

Quantum computer is a proposed mean of using quantum mechanical effects and phenomenon to achieve efficient computation and realize a new mode of information processing. When the computational model is implemented in a physical device, it must be able to adapt different internal states and provide means to perform the necessary transformation. Quantum computer is quite similar to that of a classical computer, both of which have registers. The register in quantum computer initially is set to some initial state. As the quantum computation progresses, the register will be transformed via a quantum mechanical transformation. The register may then be observed to produce the output. The difference of the quantum computer is that this machine relies on the quantum phenomenon such as interference and entanglement in the computation process, the output of any process is produced through the observation of the register. This cycle will be repeated several times before the computation is complete [6].

The power of quantum computers comes from the possibility of employing a superposition of state that allows a massive parallel data processing within one piece of quantum hardware. As a result quantum computers can efficiently solve some problems which are believed to be intractable on any classical computer [1].

A classical computer operates on strings of 0's and 1's, such as 1110010101011000, converting them into other strings. Each position in such a string is called a bit, and contains either a 0 or a 1. The only thing a step

in a classical computation can do to the state of a bit is to leave it alone, or flip it. A quantum computation can do much more to a quantum bit because quantum bits can have a much broader variety of possible states [7].

### Quantum Bits and Quantum Register

In a quantum computer, the fundamental unit of information is represented as the state of quantum subsystems, so the electric potential can be replaced by some quantum state: the quantum bit (qubit for short). Just as a bit has a state 0 or 1, a qubit also has a state  $|0\rangle$  or  $|1\rangle$ . This is called the Dirac notation and it is the standard notation for states in quantum mechanics. The difference between bits and qubits is that a qubit  $|\Psi\rangle$  can also be in a linear combination of states  $|0\rangle$  and  $|1\rangle$ :

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \dots\dots\dots (1)$$

This state is called a superposition of the states  $|0\rangle$  and  $|1\rangle$  with amplitudes  $\alpha$  and  $\beta$  ( $\alpha$  and  $\beta$  may be complex numbers). Thus, the state  $|\Psi\rangle$  is a vector in a two-dimensional complex vector space, where the states  $|0\rangle$  and  $|1\rangle$  form an orthonormal basis, called the computational basis.

To achieve any computation we need an operator to change a state to another one that will lead to make a progress in the computation process. In quantum computers these operators represent an interaction with the quantum state.

Since most unitary transformations are linear operator, therefore these operators are applied simultaneously to all the base vectors of a quantum state, thus:

$$U \sum_{i=0}^n C_i |i\rangle = \sum_{i=0}^n C_i U |i\rangle \dots\dots (2)$$

This unique feature of quantum computers is called quantum parallelism[7].

**Grover’s Algorithm**

Suppose we have an unstructured database with N elements. Without loss of generality, suppose that the elements are numbers from 0 to N – 1. The elements are not ordered. Classically, we would test each element at a time, until we hit the one searched for. This takes an

average of  $\frac{N}{2}$  attempts and N in the worst case, therefore the complexity is O (N). As we will see, using

Quantum Mechanics only  $O(\sqrt{N})$  trials are needed to achieve this task using Grover’s Algorithm. For simplicity, assume that N=2n, for some integer n. Grover’s algorithm has n qubits register. The first step is to create a superposition of all 2n computational basis states  $\{|0\rangle \dots |2n - 1\rangle\}$  of the register. Figure (1) shows the block diagram of Grover’s algorithm [9].

**Boyer, Bassared, Hoyer, Tapp (BBHT) Algorithm**

When the number of solutions is known in advance, we can use Grover algorithm to look for one of them. Without previous knowledge of the number of solutions t marked by the oracle we cannot use Grover algorithm. This impossibility arises because in the amplitude amplification process we cannot compute the number of iterations to be performed in order to maximize the coefficients of the solution. However, it is possible to use a quantum algorithm called BBHT for finding a solution in a set of items  $\{T_i\}$ ,  $i=0 \dots N-1$  given an oracle that recognizes a solution[2].

**Dürr-Hoyer Algorithm**

The Dürr-Hoyer algorithm is a quantum algorithm for finding the minimum within an unsorted table of N items. The core of the algorithm is a procedure which returns an index of items smaller than the item determined by a particular threshold index by using the BBHT algorithm. This procedure is iterated until the minimum is reached [2, 5].

**Dijkstra Algorithm**

The following algorithm presents the traditional Routing Algorithm that is based on Dijkstra’s algorithm [8]:

**Algorithm(1) The Traditional Dijkstra’s Algorithm (TDA)**

**Input** N= set of nodes of the network, S= source node, D=destination node

**Output:** The Path P<sub>D</sub>, the best path from S.

**Initialization**

**T= set of nodes so far incorporated by the algorithm.**

$W(i,j) = \begin{cases} w \geq 0 & \text{If } m \text{ and } n \text{ are directly connected} \\ \infty & \text{Otherwise} \end{cases}$

**L(n)= cost of the least cost path from s to n.**

**Add source node to set T**

**T={s}**

**Find the neighboring nodes not in T, find the compute L(n) wher  $w(s,n) \geq 0$ .**

**While  $T \neq N$  do the following :**

**find n ,  $n \notin T$  such that : L(n) is the minimum.**

**T=T U { n}**

**For each  $x$ , where  $x \in T$ , compute**  

$$L(x) = \min [ L(x), L(n)+w(n,x)]$$
  
**End while**  
**Print  $P_D$  or the best path from  $S$  to all nodes in the network.**

This algorithm consists of  $N$  iterations, where  $N$  is the number of nodes in the network, to add elements to the set  $T$  which is the set of the nodes so far incorporated by the algorithm. Each of the  $N$  iteration consists of finding the next nearest node to the source node, and then the paths of the nodes in  $T$  are recomputed according the new nearest node in a step called the updating step

For each  $x$ , where  $x \in T$ , compute  $L(x) = \min [ L(x), L(n)+w(n,x)]$ .

Quantum Dijkstra Algorithm Dijkstra algorithm is the one that is widely used in the internet routing, has been improved by adding some of the concepts of quantum computing to improve the time complexity and execution time of some steps within it.

This work presents a suggestion for an algorithm that applies Dürr-Hoyer algorithm to the step find  $n$ ,  $n \notin T$  such that :  $L(n)$  is the minimum.in algorithm (1) to find the next nearest node to the source node, here after it is called the Quantum Dijkstra algorithm (QDA).

The time complexity of the step find  $n$ ,  $n \notin T$  such that :  $L(n)$  is the minimum. in the traditional Dijkstra’s algorithm is  $O(N)$ , where  $N$  is the number of involved nodes, while the same step after using the quantum approach will be  $O(\sqrt{N})$ , although the time complexity of the entire algorithm is still  $O(N^2)$  because of the updating step.

### The DH in QDA

The Dürr-Hoyer algorithm is used here to find the neighboring node not in  $T$  in the algorithm QDA that has the least cost path from node  $s$  (or the nearest node to  $s$  in  $T$ ) to incorporate it into  $T$ . This step uses the details of Dürr-Hoyer algorithm such as the quantum registers, the oracle function, the number of iterations and the diffusion function will be shown in the next sections.

### The Quantum Registers in QDA

In the beginning of applying the Dürr-Hoyer in QDA, all the nodes that are not in  $T$  and connected to the least cost node in  $T$  that are updated from the preceding time step, are stored in the input register with their cost value. The first one of these nodes in input register is stored in the output register, which at the end of the algorithm will contain the least cost node.

### Register Initialization and Hadmard Transform

At the beginning of the algorithm the amplitude field of the input register is initialized to the grounded state configuration  $|0\rangle_N$  where  $N$  is the number of nodes in the input register.

Then, the Hd transform of this field distributes the same amplitude to each node in this register before the application of the oracle function.

After the applications of the Hd transform the input register will be as in equation (4.5).

### The Oracle Function for QDA

The oracle function is that the amplitude of the node which has smaller cost value than the cost of the node in

the output register is marked with minus sign. Therefore the oracle function is as follows:

$$R|i\rangle = \begin{cases} 1 & \text{if } \text{cost}(|i\rangle) \geq \text{cost}(|i_0\rangle) \\ \dots(3) \\ -1 & \text{otherwise} \end{cases}$$

Where  $|i\rangle$  is any sub state in the input register and  $|i_0\rangle$  is the state of the node in the output register.

At the end of this step all nodes that have cost value less than the node in the output register will be with negative amplitude.

**The Diffusion Function for QDA**

This function is used to increase the amplitude of the items that are marked with the minus sign because of the application of the oracle function and decrease in the amplitude of those which are not marked, by the traditional diffusion equation of Grover’s algorithm.

$$|\Psi_k\rangle = \sin\left[\left(\frac{2 * k + 1}{2}\right) * \theta\right] |u\rangle + \cos\left[\left(\frac{2 * k + 1}{2}\right) * \theta\right] |i_0\rangle \dots\dots(4)$$

Where k is the iteration and  $\theta$  is the rotation angle.

**The Rotation Angle and the Number of Iterations**

The rotation angle is computed according to the number of nodes in the input register since this algorithm considers that only one solution at each iteration (even there are more solutions). Then, if the state of the input register  $|\Psi\rangle$  is:

$$|\Psi\rangle = \frac{\sqrt{N-1}}{\sqrt{N}} |u\rangle + \frac{1}{\sqrt{N}} |i\rangle \dots\dots(5)$$

then the angle  $\theta$  is computed as follows:

$$\theta = 2 \arccos \frac{\sqrt{N-1}}{\sqrt{N}} \dots\dots(6)$$

the number of iterations is computed to be smaller than or equal to a maximum number of iterations m which must be determined in the beginning of this algorithm. The value of m must be a small value in its initialization (in most cases its being 1 or 2).

Sometimes the increment of the amplitude by the amplitude amplification operators is not enough to distinguish the solution nodes from the others. Therefore, increment in the maximum number of iteration is required. So, m is increased by multiplying it with a factor value (also it should be determined and adjusted previously according to the application). In this algorithm the value  $\lambda=6/5$  is used as in the original algorithm.

**Algorithm (2) QDA Algorithm**

**Input** N= set of nodes of the network, S= source node, D=destination node

**Output:** The Path P<sub>D</sub>, the best path from S.

**Initialization**

T= set of nodes so far incorporated by the algorithm.

$W(i,j) = \begin{cases} w \geq 0 & \text{If } m \text{ and } n \text{ are directly connected} \\ \infty & \text{Otherwise} \end{cases}$

**L(n)= cost of the least cost path from s to n.**

**Add source node to set T**

T={s}

**Find the neighboring nodes are not in T, find the compute L(n) where  $w(s,n) \geq 0$ .**

**While  $T \neq N$  do the following:**

```

/* the following loop is substitute the step
:
    Find n , n ∉ T such that : L(n) is the
    minimum.
In the TDA */
    Repeat
Put the nodes P=(p1,...,pN), connected to the
nearest node in T and updated in the last
time step in the input quantum register and
p1 in the output register.
    Apply H transform
Node(pi).amplitude= 1/√N
Compute the rotation angle θ = 2 * √(N-1) / √N
Compute the number of iterations
k = (π - θ) / 2θ
Apply the Oracle Function to sign with
minus the node pj,
    that satisfies the following:
    f(pi) = { -1 if pi.name = pj.name, and pi.cost < pj.cost
             1 otherwise
Apply the diffusion function
|Ψk⟩ = sin[ ((2*k+1)/2) * θ ] |u⟩ + cos[ ((2*k+1)/2) * θ ] |i0⟩
Measure Input-Register to find a solutions
X
Put X in Output-Register
Until X = ∅
    
```

```

T=T U { n}.
For each x , where x ∈ T , compute
L(x) = min [ L(x), L(n)+w(n,x)]
Print Pd or the best path from S to all
nodes in the network.
    
```

### Experimental Results and Conclusion

In this work the suggested algorithm is simulated on a traditional computer. Figure (3) shows the relation between the number of nodes in the network and the number of iterations at the step find n , n ∉ T such that : L(n) is the minimum. of the TDA and substitution in QDA, in the TDA the number of iterations is the same as the number of nodes while in QDA the average number of iteration is half the number of nodes.

The time complexity of both TDA and QDA is the same O(N<sup>2</sup>) , but the execution time of QDA can be reduced if the quantum part of this algorithm is executed on a quantum computer.

The time complexity of the simulation of a quantum algorithm is larger than any other algorithm that is implemented traditionally. To gain the required efficiency of the quantum algorithms requires to implementing these algorithms on a quantum computer instead of a traditional computer.

### References

[1].Chi, D. P., Kim, J., “Quantum Computation”, Technical paper, Department of Mathematics, Seoul National University, 1997.

- [2].Durr, C., Hoyer, P., “A Quantum algorithm for Finding the Minimum”, *quanta-ph/9607014 V2*, 1999.
- [3].Faraj, S.T., “Quantum Cryptographic Key Distribution in Optical Communication Network”, Ph.D. Thesis, Nahrain University, Iraq, 1999.
- [4].Huston, G., “An Introduction to IP routing, Technical Report, [www.potaroo.com](http://www.potaroo.com),2004.
- [5].Malossini, A., Blanzieri, E, E., Calarcot, “QGA: A Quantum Genetic Algorithm”, Technical Report, University of Trento, Department of Information and Communication Technology, 2004.
- [6].Mc Cubbin, C. B., “Open QUACS, an Open Source Quantum Computation Simulator in Maple”, MSc. Thesis, Faculty of Graduate School, 2000.
- [7].Mermin, N. D., “Lecture Notes on Quantum Computation and Information Theory”, Cornell University, Physics 481-681, CS 483, Fall, 2000.
- [8].Stallings, W., *Computer Networking with Internet Protocols and Technology*, Pearson-Education Pte Ltd Indian, 2004.
- [9].Viamontes, G.F., Markov, I. L., Hayes, J. P., “Is Quantum Search Practical ?”, *IEEE Computing in Science and Engineering*, pp. 22-30, May/June 2005.

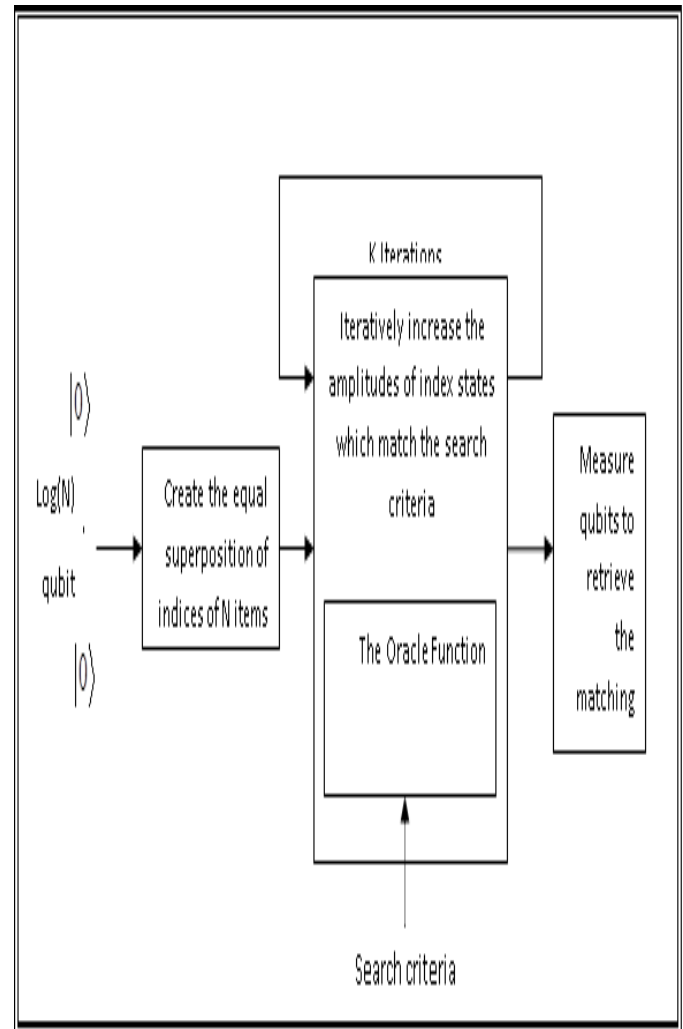


Figure (1) the block diagram of Grover's algorithm

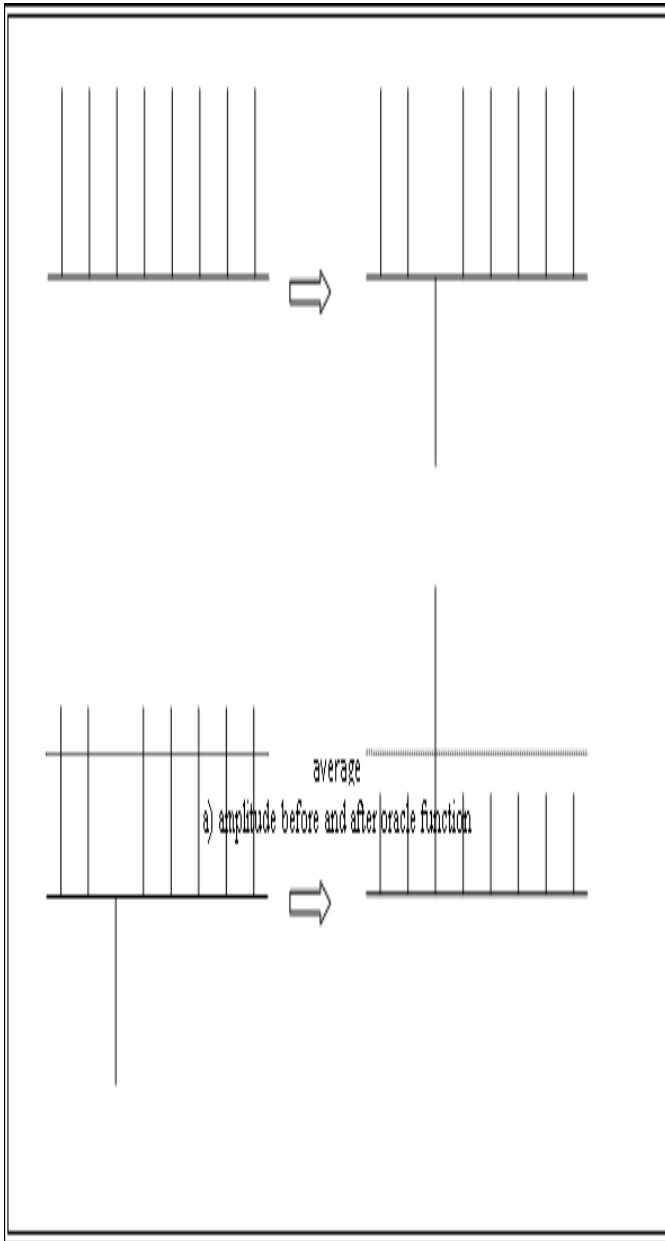


Figure (2) The effect of oracle and diffusion operators of Grover's Algorithm

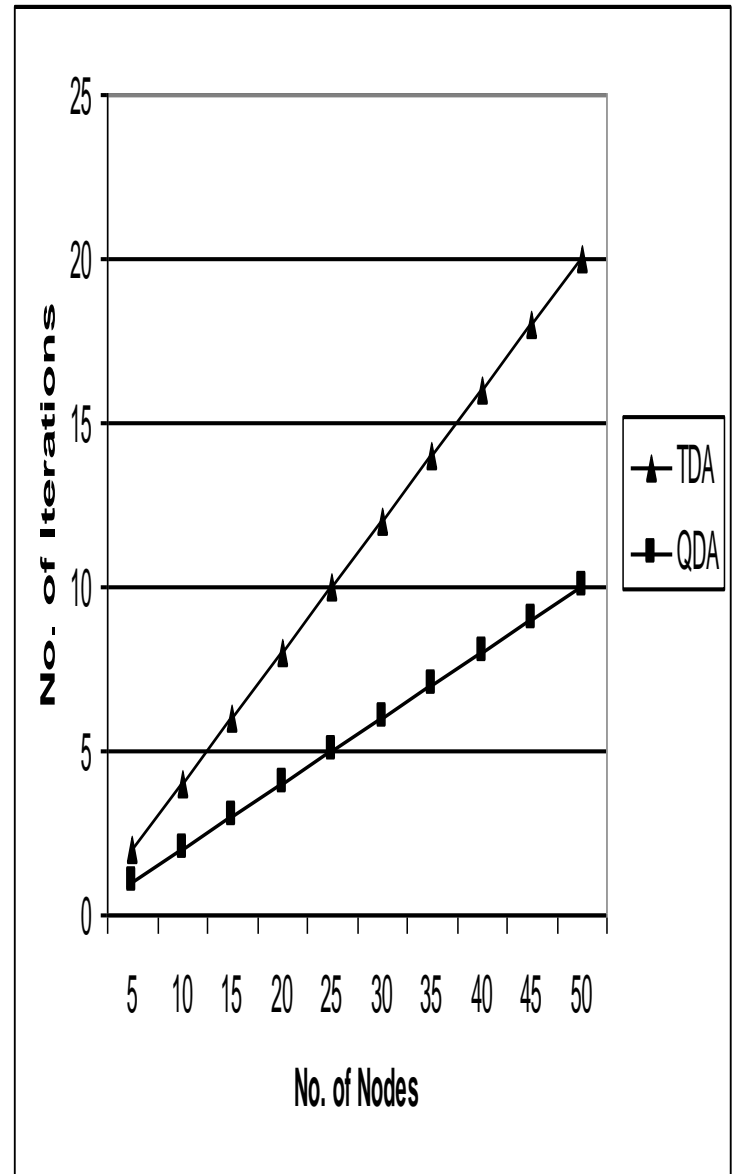


Figure (3) A comparison between TDA and QDA



## تطوير كمي لخوارزمية Dijkstra لتحديد المسار في شبكات الحاسبات

رنا فريد غني

احمد طارق صادق

### الخلاصة

الهدف من هذا البحث هو تطوير خوارزمية Dijkstra والتي تستخدم على نطاق واسع في تحديد مسار المعلومات في شبكات الحاسبات. استخدام الاحتساب الكمي في تطوير خوارزمية Dijkstra يتم عن طريق استثمار الامكانية العالية للمعالجة المتوازية الموجودة في هذه الطريقة للاحتساب في البيئة الكمية للتعامل مع المتطلبات المتزايدة الناتجة عن التزايد في حجم شبكو الانترنت.

الخوارزمية المقترحة تمت مقارنتها مع خوارزمية Dijkstra من ناحية عدد الدورات والتعقيد الزمني والنتائج اثبتت افضلية الخوارزمية الكمية.