**International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)**
**Web Site: www.ijettcs.org Email: editor@ijettcs.org**
**Volume 4, Issue 5(2), September - October 2015** **ISSN 2278-6856**

# Shortest Path through Grover's Algorithm in MATLAB

**Nidhi Jain[1], Dr. Mahesh Porwal[2]**

[1]M. Tech. Student Department of Electronics and Communication Engineering
Shrinathji Institute of Technology and Engineering, Nathdwara (Rajasthan)

[2]Professor Department of Electronics and Communication Engineering
Shrinathji Institute of Technology and Engineering, Nathdwara (Rajasthan)

## Abstract

***Classical algorithms have been used to search over some space for finding the shortest paths problem between two points in a network and a minimal weight spanning tree for routing. Any classical algorithm deterministic or probabilistic will clearly used O(N) steps since on the average it will measure a large fraction of N records. Quantum algorithm is the fastest possible algorithm that can do several operations simultaneously due to their wave like properties. This wave gives an $O(\sqrt{N})$ steps quantum algorithm for identifying that record, where was used classical Dijkstra's algorithm for finding shortest path problem in the graph of network and implement quantum search. Also we proposed the structure for non-classical algorithms and design the various phases of the probabilistic quantum classical algorithm for classical and quantum parts. Finally, we represent the result of implementing and simulating Dijkstra's algorithm as the probabilistic quantum-classical algorithm.***

**Keywords:** Graph Theory, Algorithm Design, Quantum Algorithm, Network Routing

## 1. INTRODUCTION

The quantum search algorithm performs a generic search for a solution to a very wide range of problems. Quantum searching is a tool for speeding up these sorts of generic searches through a space of potential solutions. The problem of unstructured search is paradigmatic for any problem where an optimal solution needs to be found in a black box fashion, i.e., without using the possible structure of the problem:

Assuming there is a system with $N = 2^n$ states labeled $S_1,S_2,..,S_N$.These $2^n$ states are represented by $n$ bit strings. Assuming there is a unique marked element $S_m$ that satisfies a condition $C(S_m) = 1$, and for all other states $C(S) = 0$. Suppose that C can be evaluated in unit time. The task is to devise an algorithm which minimizes the number of evaluations of C. The idea of Grover's algorithm is to place the register in an equal superposition of all states, and then selectively invert the phase of the marked state, and then perform an inversion about average operation a number of times. The selective inversion of the marked state followed by the inversion about average steps has the effect of increasing the amplitude of the marked state by $O(1/\sqrt{NO(1)})$. Therefore after $O\sqrt{N}$ operations the probability of measuring the marked state approaches 1 (Grover, 1996). [1][2][3]

Grover's algorithm is as follows:

• Prepare a quantum register to be normalized and uniquely in the first state. Then place the register in an equal superposition of all states $(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \dots \frac{1}{\sqrt{N}})$ by applying the Walsh-Hadamard operator W . This means simply the state vector will be in an equal superposition of each state.

• Repeat $O\sqrt{N}$ times the following two steps (the precise number of iterations is important, and discussed below):

o Let the system be in any state S. If $C(S) = 1$, rotate the phase by π radians, else leave system unaltered. It is worth noting that this operation has no classical analog. One cannot observe the state of the quantum register, doing so would collapse the superposition. The selective phase rotation gate would be a quantum mechanical operator which would rotate only the

o amplitude proportional to the marked state within the superposition.

o Apply the inversion about average operator A, whose matrix representation is: $A_{ij} = 2/N$ if $i \neq j$ and $A_{ij} = -1+2/N$ to the quantum register.

• Measure the quantum register. The measurement will yield the n bit label of the marked state $C(S_M) = 1$ with probability at least 1/2 (Grover, 1996).[4][5]

This Grover's algorithm flow chart is as shown in Figure 1.

## 2. PROPOSED SYSTEM

The proposed system is concerned with the simulation of Grover's Algorithm using MATLAB. Quantum computing uses unitary operators acting on discrete state vectors. Matlab is a well-known (classical) matrix computing environment, which makes it well suited for simulating quantum algorithms. The number of database elements is $2^4 = 16$. The desired element is randomly generated from among the 16 elements using Matlab's *rand* function and it is the 13th element. The quantum gates are defined using Matlab's *eye* function. The optimal number of iterations is determined by $(\pi/4)\sqrt{N}$ as proposed by Grover. Grover demonstrated that quantum computers can perform database search faster than their classical counterparts. In this simple example of Grover's algorithm, a haystack function is used to represent the database[7][8]. We are searching for a needle in the haystack, i.e. there is one element of the database that we require. Grover's algorithm works by iteratively applying the inversion about the average operator to the current state. Each iteration amplifies the probability of a measurement $(\pi/4)\sqrt{N}$ collapsing the state to the correct needle value. Grover showed that performing a measurement after iterations is highly likely to give the correct result.

The Walsh-Hadamard transformation, operator to rotate phase and inversion about average transformation were achieved through matrices of zeros and ones available as Matlab commands. At the end of the program, the output was set to be a movie-like display of the different stages of the iteration in a 3-dimensional plane comprising the axes Amplitude, States and Time. This was achieved by using Matlab's function movie[9].

## 3. OPTIMALITY OF GROVER'S ALGORITHM

It is stated in Grover (1996) that his result was optimal, but it is not directly proved. Grover's algorithm takes $O(\sqrt{n})$ iterations, and is thus asymptotically optimal. It has been shown since that any quantum algorithm would require at least $\pi/4\sqrt{N}$ queries, which is precisely the number queries required by Grover's algorithm (Grover, 1999)[10].

## 4. RESULT

We implemented and simulated the Grover's algorithm with Matlab on classical computer. We have tested this algorithm with $N=2^n$ possible inputs that n is number of qubits. The simulation results for n=4 qubits as a data index is shown in the figure 4. In these diagrams, number of possible inputs is N=16 and this number is length of queue Q. We assumed that there is one solution in queue Q. The amplitude value of solution in Grover's algorithm reaches to 1 after $(\pi/4)\text{Sqrt}(16)=3.14$ iterates and the amplitude value of other data reached to zero. Figure 2 show that with 3 iteration we can find solution in queue

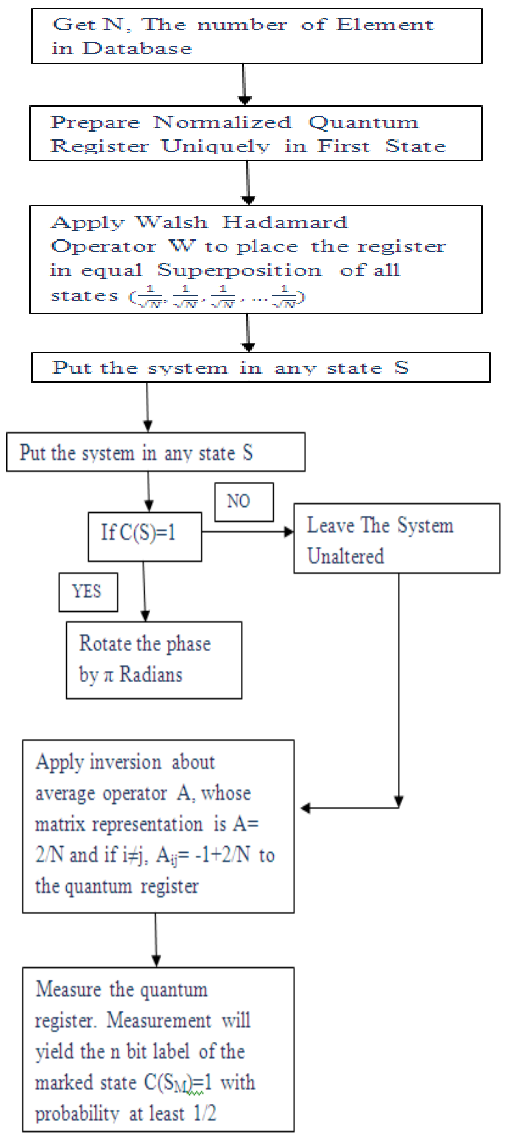

Get N, The number of Element in Database
Prepare Normalized Quantum Register Uniquely in First State
Apply Walsh Hadamard Operator W to place the register in equal Superposition of all states $(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots \frac{1}{\sqrt{N}})$
Put the system in any state S
Put the system in any state S
If C(S)=1
NO
Leave The System Unaltered
YES
Rotate the phase by π Radians
Apply inversion about average operator A, whose matrix representation is A= 2/N and if i≠j, $A_{ij}$= -1+2/N to the quantum register
Measure the quantum register. Measurement will yield the n bit label of the marked state $C(S_M)=1$ with probability at least 1/2

**Figure 1** Flowchart of Grover Algorithm

Q. The maximum iteration of algorithm is $(\pi/4)\sqrt{N}$. we compare the speeds of Dijkstra's algorithm in three states of implementation for finding the shortest path in graph. These states depend on implementation of EXTRACT_MIN procedure as a linear array, or as a binary heap, or as a quantum search. When a *Q* is implemented as a linear heap or quantum search, the algorithm is more speed up than as a linear array[11][12].

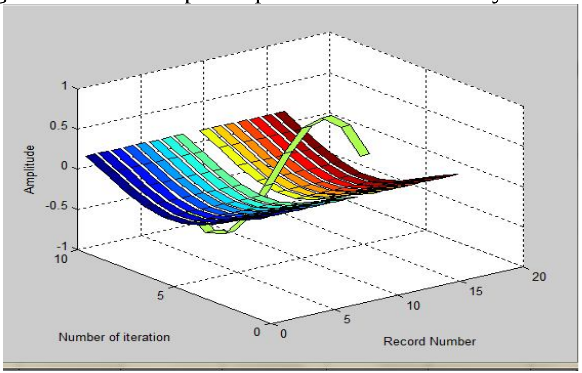
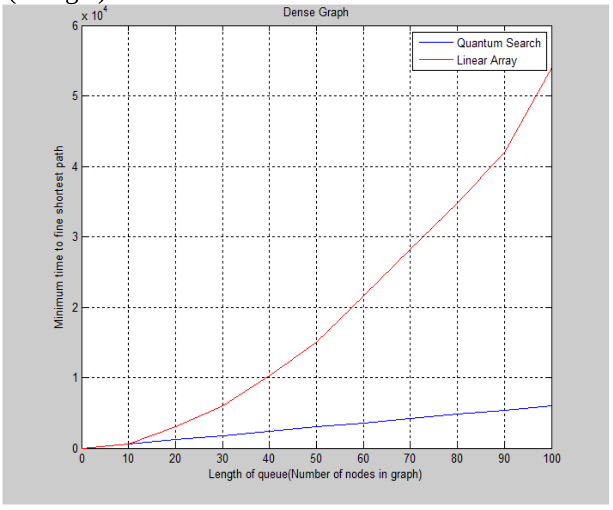


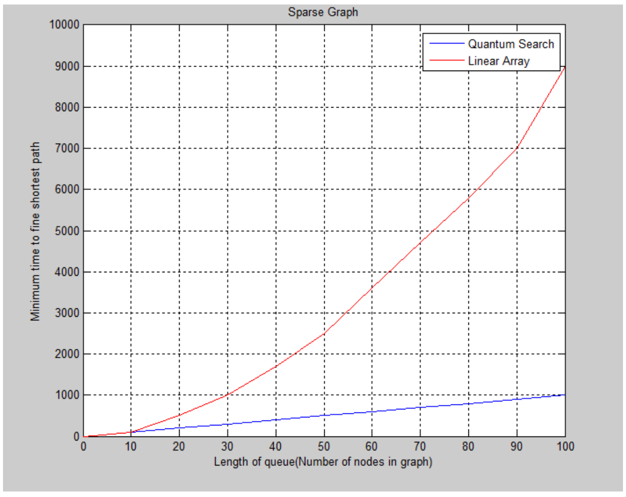
**Figure 2** Amplitude Value. The result of quantum search algorithm with 4 qubits input data and 16 elements in queue with 3 iterations

## 5. CONCLUSION

With comparing the time complexities of the versions of Grover's algorithms, we can see that the time taken by Grover's algorithm is determined by the speed of the queue operations. When a *Q* is implemented as a linear array, EXTRACT_MIN takes O(V) time and there are |V| such operations. Hence, the running time of the algorithm with array implementation is O(V2 + E) = O(V2). When a *Q* is implemented as a binary heap, EXTRACT_MIN operations takes O(lg V) time and there are |V| such operations. Hence, the running time of the algorithm with binary heap provided given graph is O((V + E) lg V). Note that this time becomes O((V+V)logV)=O(ElogV) if all vertices in the graph is reachable from the source vertices and the graph is sparse. If graph be dense, the running time of the algorithm is $O((V+V^2)\log V) = O(V^2 \log V)$[13][14].



(a) Dense Graph



(b) Sparse Graph

**Figure 3** Speeds comparison of Dijkstra's algorithm in three states of implementation to find the shortest path in the graph.

## REFERENCES

[1]. Michael A. Nielsen & Isaac L. Chuang, Quantum Computation and Quantum information, 10th anniversary edition, Cambridge University press.
[2]. Phillp Kaye, Raymond Laflamme, Michele Mosca, An Introduction to Quantum Computing, Oxford University Press
[3]. Bennett C.H, Bernstein E., Brassard G. and Vazirani U. (1996). Strengths and Weaknesses of Quantum Computing. In the proceedings of SIAM Journal of Computing.
[4]. Boyer M., Brassard G., Hoyer P. and Tapp A., Tight Bounds on Quantum Searching. In the Proceedings of PhysComp. (lanl e-print quant-ph/9701001).
[5]. Deutsch, D., and Jozsa, R.(1992). Rapid Solutions of Problems by Quantum Computation, Proceedings of the Royal Soc. of London, 439, 553.
[6]. Eppstein D., Irani S., and Dillencthet M. (2000). ICS 260-Fall Class Notes 11: Turing Machines, Non-determinism, P and NP. Available at: http://www1.ics.uci.edu/~eppstein/260/notes/notes11.ps. Accessed on 20th April, 2010.
[7]. Grover, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. In Proceedings of 28th Annual ACM Symposium on the Theory of Computing, New York, pp. 212.
[8]. Papadimitriou, C (1994). Computational Complexity, Addison-Onesley Publishing Company.
[9]. Shor, P. W. (1994). Algorithms for Quantum Computation: Discrete Logs and Factoring. In Proc. 35th Annual Symposium on Foundations of Computer Science.
[10]. Williams C. and Clearwater S. (1998). Explorations in Quantum Computing, Springer-Verlag, New York, Inc.
[11]. Mathlab Implementation of Quantum Computation in Searching an Unstructured Database , I.O. Awoyelu and P. Okoh , Department of Computer Science & Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria.

## AUTHOR

**Nidhi Jain** received the B. Tech. degree in Electronics and Communication Engineering from Geetanjali Institute of Technical Studies in 2011 and pursuing M. Tech. in ECE with specialization in Digital Communication from Shreenathji Institute of Technology and Engineering.