Journal of Computer Science 5 (10): 725-731, 2009
ISSN 1549-3636
© 2009 Science Publications

# A Hybrid Architecture Approach for Quantum Algorithms

[1]Mohammad Reza Soltan Aghaei, [2]Zuriati Ahmad Zukarnain,
[3]Ali Mamat and [4]Hishamuddin Zainuddin
[1]Department of Communication Technology and Network,
Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia
[2]Department of Computer Science, University Putra Malaysia, Malaysia
[3]Institute for Mathematical Research, University Putra Malaysia, Malaysia

**Abstract: Problem statement:** In this study, a general plan of hybrid architecture for quantum algorithms is proposed. **Approach:** Analysis of the quantum algorithms shows that these algorithms were hybrid with two parts. First, the relationship of classical and quantum parts of the hybrid algorithms was extracted. Then a general plan of hybrid structure was designed. **Results:** This plan was illustrated the hybrid architecture and the relationship of classical and quantum parts of the algorithms. This general plan was used to increase implementation performance of quantum algorithms. **Conclusion/Recommendations:** Moreover, simulation results of quantum algorithms on the hybrid architecture proved that quantum algorithms can be implemented on the general plan as well.

**Key words:** Quantum algorithm, quantum computing, hybrid architecture

## INTRODUCTION

A quantum computer is a device that takes advantage of quantum mechanical effects, such as superposition and entanglement, to perform certain computations faster than a purely classical machine can. Quantum parallelism is best understood in the context of the concepts of superposition, entanglement and measurement. If large-scale quantum computers can be built, they will be able to solve certain problems much faster than any of our current classical computers. Quantum computers are different from traditional computers based on transistors. Some computing architectures such as optical computers may use classical superposition of electromagnetic waves. Without some specifically quantum mechanical resources such as entanglement, it is conjectured that an exponential advantage over classical computers is not possible.

Today, many researchers are doing research to design a quantum computer with the implementation of quantum algorithms. A lot of work has been done over the last decade, since the Shor's discovery of quantum discrete logarithm and factoring algorithms[6] and Grover's publication of a quantum search algorithm[7]. Nonetheless, these algorithms still stand as far and away the most significant developments in quantum computation. The performance of quantum algorithm can be evaluated in terms of speed, efficiency and implementation of quantum circuits.

The objective of our research is undertaken to analyze and design a general plan of the hybrid architecture for quantum algorithms. The plan is being used to increase performance of implementation of quantum algorithms. For finding a general plan, we must analyze and compare the existing quantum algorithms. Moreover, the hybrid architecture will be designed for quantum algorithms and the relationship of classical part and quantum part of algorithms will be extracted. Finally we verify and simulate the existing quantum algorithms with this plan and represent the results of implementation and simulation of quantum algorithms.

**Related study:** A primary goal of the theory of quantum complexity is to determine when quantum computers may offer a computational speed-up over classical computers. At present, there are only a few general techniques known in the field of quantum computing and finding new problems which give a polynomial time quantum algorithm for some problem for which no classical polynomial time solution is

**Corresponding Author:** Mohammad Reza Soltan Aghaei, Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia

known. Given the possible power of quantum parallelism, much work has been done to show formally with mathematical proofs how quantum computers differ from classical ones in their power to compute things. There are few quantum algorithms that can be implemented on quantum computer to solve certain problems. In the next, the quantum algorithms will be analyzed to find a general plan of quantum algorithms. The plan shows the hybrid architecture of quantum algorithms.

In 1980 Paul Benioff offered a classical Turing machine which used quantum mechanics in its workings, thus showing that theoretically a quantum computer was at least as powerful as a classical computer[1].

The first quantum algorithm is Deutsch's algorithm[2,3] which can determine whether a function is constant ($f(0) = f(1)$) or balanced ($f(0) \neq f(1)$), using only a single call to the function. Note that classically, to solve this problem with a success probability bigger than one half, a machine has to query the black box twice; both f(0) and f(1) are needed. Deutsch's ingenuity is to use interference of the amplitudes of the quantum state such that only one query to the black box suffices. The following circuit on two qubits gives the quantum algorithm. As a result, Deutsch's algorithm saves one query in comparison to the best possible classical algorithm for this problem. One query might seem very little, yet we will see how this algorithm has been generalized in several steps to ultimately factor numbers.

Deutsch and Jozsa[4] showed in a research in 1992 that there was an algorithm that could be run in poly-log time on a quantum computer, but required linear time on a deterministic Turing machine. This may have been the first example of a quantum computer being shown to be exponentially faster than a deterministic Turing machine. Unfortunately, for the quantum computer, the problem could also be solved in poly-log time in a probabilistic Turing machine, a Turing machine which is capable of making a random choice. The Deutsch-Jozsa algorithm can determine whether a function f that maps n bits to one bit, is constant or balanced, using only a single call to the function. Note that classically, to solve this problem deterministically, one needs $2^{n-1}+1$ queries in the worst case. The Deutsch-Jozsa algorithm solves this problem with one quantum query with the following algorithm.

The algorithm of Simon[5] finds the "period" of a function. This algorithm finds the hidden string s in Simon's Problem. Simon's problem requires an exponential number of queries on a classical computer. One can show that the best any classical probabilistic machine can do is to query elements at random until a collision is found. The probability of a collision for two randomly chosen elements is about $2^{-n}$ and a slightly more elaborate analysis shows that the expected number of queries until a collision happens among the queried elements is $O(2^{n/2})$. The expected number of evaluations of the function in the execution of the algorithm is less than n and the expected number of other elementary gates is in $O(n^3)$.

Shor's algorithm is a quantum algorithm for factoring an integer N in $O((\log N)^3)$ time and O(log N) space. A common public-key cryptography method known as RSA is based on the assumption that it is computationally infeasible to factor a large integer. For this reason a quantum computer with sufficiently many quantum bits could "break" RSA. RSA uses a public key N which is the product of two large prime numbers. One way to crack RSA encryption is by factoring N, but with classical algorithms, factoring becomes increasingly time-consuming as N grows large; more specifically, no classical algorithm is known that can factor in time which is polynomial in log N. The first necessary observation is that in order to find a factor of a number, it is sufficient to solve a problem called period finding, the problem Shor's algorithm[6].

The Grover's algorithm performs a generic search for a solution to a very wide range of problems. Consider any problem where one can efficiently recognize a good solution and wishes to search through a list of potential solutions in order to find a good one[7-9]. 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. Classically, a deterministic algorithm needs to make $2^n-1$ queries to identify w in the worst case and a probabilistic algorithm still needs $O(2^n)$ queries. Grover gave a quantum algorithm that solves this problem with $O(2^{n/2})$ queries and this is known to be the best possible. Grover's algorithm can hence speed up any algorithm that uses searching as a subroutine.

## MATERIALS AND METHODS

In this study, first a general plan of the hybrid architecture will be designed for quantum algorithms and also the relationship of classical part and quantum part of algorithms will be extracted. We determine the complete cycle of the hybrid architecture for the quantum algorithms and verify the existing quantum algorithms with this plan. These algorithms according

to the hybrid architecture are simulated by MATLAB. Finally, the results of implementation and simulation of the hybrid architecture for Grover's search algorithm and Shor's period finding algorithm are represented. The explanations of other quantum algorithms are same.

**A hybrid architecture for quantum algorithms:** The gate-based quantum computers are not universal. On the other hand, global unitary operations like the shift operator cannot be expressed within the circuit model, cannot be equally applied to machines with unlimited and limited memory and cannot be assumed to be equally available on different quantum hardware architectures.

To overcome the above restrictions, quantum programming uses a classical universal language to define the actual sequence of elementary instructions for a quantum computer, so a program is not intended to run on a quantum computer itself, but on a (probabilistic) classical computer, which in turn controls a quantum computer and processes the results of measurements. In the terms of classical computer science, you can describe this setting as a universal computer with a quantum oracle. Figure 1 shows this hybrid architecture.

The quantum algorithms such as Shor's algorithm consists of two parts; first part is classical algorithm which can be done on a classical computer and second part is Quantum algorithms which can be done on a quantum computer or simulate on classical computer[10-12]. In this study, a general plan is illustrated to show the hybrid architecture and to find the relationship of classical and quantum parts of the algorithms. Figure 2 shows the relationship of classical part and quantum part of the algorithm.

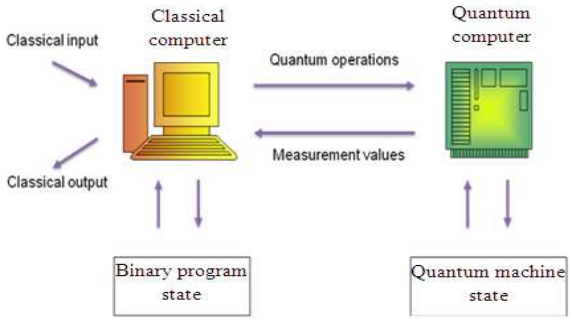


Fig. 1: The hybrid architecture between classical and quantum computers

Naturally, the quantum algorithms are the hybrid algorithms that consist of classical and quantum components. Moreover, the quantum portion of many algorithms is probabilistic; often need multiple runs to get the desired result. The complete cycle of the hybrid architecture for the quantum algorithms will be done as follows:

1. Pre-calculate certain classical factors (initialize and run the classical part of the algorithm)
2. Running the quantum algorithm by the quantum circuit
   a. Initialize the quantum node (Initialize quantum circuit and define all gates, switches and unitary function)
   b. Prepare inputs state (store inputs on target and control registers)
   c. Execute the quantum portion of the algorithm (Apply gates and unitary transformation on input data)
   d. Measure the output of Machine State (Measure the output registers of the quantum circuit)
   e. Evaluate Measurement (If have the desired result, then doing post-processing in step 3)
   f. Exit if desired result (If solution found then exit from quantum circuit, else repeat step 2)
3. Finish post-processing (Run the second classical part of the algorithm)

Steps 1 and 3 been executed on classical computer and step 2 been executed on quantum computer by quantum circuits. Measuring and evaluating of the quantum circuit in steps 2(e) and 2(f) can be done on classical computer. The diagram in Fig. 3 shoes the development of a general plan of hybrid architecture for the quantum algorithms and being simulated on classical computer. The quantum circuit is simulated on classical computer.

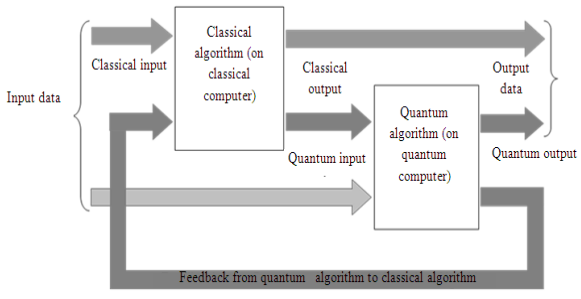


Fig. 2: The relationship of classical part and quantum part of the hybrid algorithm

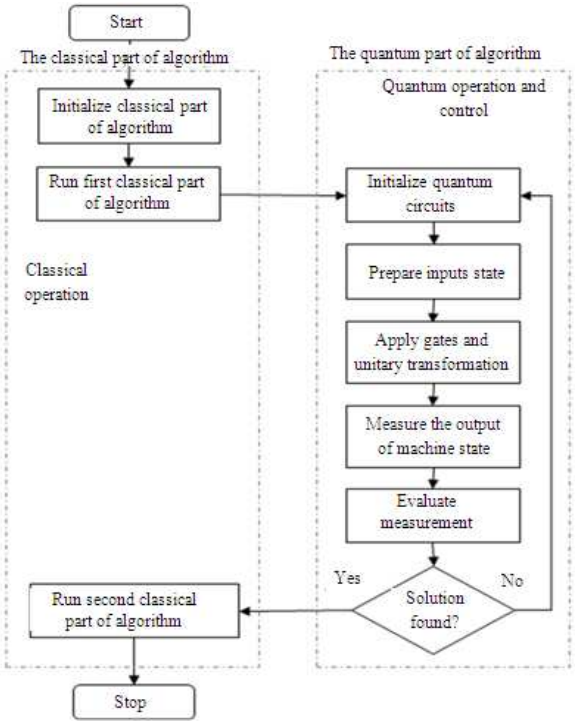


Fig. 3: The classical part and the quantum part of the hybrid algorithm

## RESULTS

A general plan of hybrid architecture for the quantum algorithms can be developed and implemented for circuits of the quantum algorithms. This plan is useful for simulation and architecture design of quantum computer to be run quantum algorithms and inspired a development of new quantum algorithm. The circuits of the quantum algorithms are simulated using MATLAB based on the hybrid architecture.

In its simplest form, a quantum algorithm is consisting of a unitary transformation and a subsequent measurement of the resulting state. The common part in all quantum algorithms is the black box or oracle function $U_f$ that is often used to model a subroutine of calculates and is reversible. Classically, a black-box function can be simply thought of as a box that evaluates an unknown function f. A black box function is often used to model a subroutine of calculate. Many of the separations between classical and quantum computing power will be formulated in the black box or oracle model. For certain problems a quantum algorithm needs to make substantially less calls or queries to the black box than any classical algorithm. Classically, a black-box function can be simply thought of as a box that evaluates an unknown function f. The input is some n-bit string $|x\rangle$ and the output is given by an m-bit string f(x). To create a reversible box, the input $|x\rangle$ is output together with f(x). This reversible box, when given to a classical machine, is no stronger than the corresponding simple non-reversible box that maps x to f(x). Note that this box now induces a transformation on n+m-bit strings that can be described by a permutation of the $2^{n+m}$ possible strings; in particular it is unitary.

At the end of the calculation in the quantum part of hybrid algorithm, the result register is a superposition of all of the results, one for each of the $2^n$ possible inputs. However, we can't directly read out all of those results. If we measure the result register to get our answer, the superposition collapses into a single state with a probability according to the weights discussed above. Then we have only a single value; our end result is no better than if we had used a classical computer to compute the function for one possible input chosen at random. The measurement of a qubit causes the collapse of the wave function, forcing the state of the system into just one term of the superposition.

In this study the initialization and setup of the existing quantum algorithms have been demonstrated based on the hybrid architecture. Moreover, the well-known quantum algorithms are implemented and simulated on this hybrid architecture.

According to the classical-quantum algorithm, the complete cycle of the hybrid architecture for classical and quantum part of the algorithm will be done. In the classical part, pre-calculate certain classical factors conduct and then the quantum part will be started. In the quantum part, two different inputs $|x\rangle$ and $|y\rangle$ is initialized depend on quantum algorithm. The input $|x\rangle$ is stored in the control register and second input $|y\rangle$ in the target register. In the next, as the circuit of quantum algorithm, Hadamard or QFT gates are applied on target and control registers sequentially. Moreover, the unitary function $U_f$ is applied on two registers and then, inverse of the input gate is applied on target register. Finally the output is measured and the result is evaluated. The classical post processing of the hybrid algorithm is run, if we achieve the desired result, otherwise the quantum algorithm iterates. A feedback is needed to ensure the iteration of the quantum algorithm.

The simulation results of quantum search algorithm is shown in the Fig. 4 for n = 6 qubits as a data index. In these diagrams, number of possible inputs is N = 64 and this number is length of data queue. We assumed that there is one solution. The amplitude value of solution in Grover's algorithm reaches to 1 after $(\pi/4)$ Sqrt (64) = 6.28 iterates and the amplitude value of

other data reached to zero. Fig. 4 shows that with 6 iterate we find solution and if we continue to run the algorithm, then the amplitude value of solution keep a way from 1 and we lose solution. The maximum iteration of algorithm is $(\pi/4)\sqrt{N}$.

The simulation results for n = 12 qubits as a data index will be shown in the Fig. 5. In these diagrams, number of possible inputs is N = 4096 and this number is length of the data queue. The element of 3750 is desired key. The amplitude value of solution in Grover's algorithm reaches to one after $(\pi/4)\sqrt{N} = 50$ iterates and the amplitude value of other data reached to zero. In the next section, the results of implementation and simulation of the complete cycle of Grover's search algorithm and Shor's period-finding algorithm based on the hybrid architecture are represented.

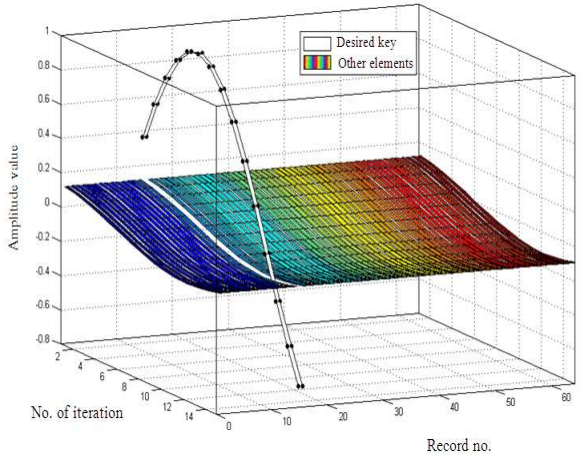


Fig. 4: The result of quantum search algorithm with 6 qubits input data and 64 elements with 15 iterations

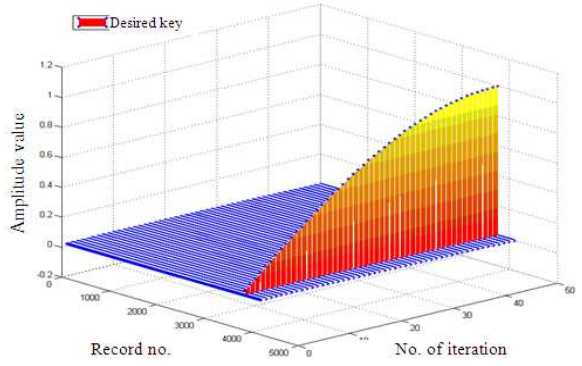


Fig. 5: The simulation result of quantum search algorithm with 12 qubits input data and 4096 elements in queue with $(\pi/4)\sqrt{N}$ = 50 iteration

## DISCUSSION

In this study the initialization and setup of the existing quantum algorithms have been demonstrated based on the hybrid architecture. Moreover, the well-known quantum algorithms are implemented based on the hybrid architecture and simulated by MATLAB. However, the explanation of setting up quantum search algorithm and order-finding problem on the hybrid architecture seems to be adequate.

Quantum search algorithm can hence speed up any algorithm that uses searching as a subroutine. Stages of the algorithm distribute to classical and quantum parts. In the classical part of algorithm, pre-calculate certain classical factors conduct and then the quantum part will be started. In the quantum part, the n-qubit input $|x\rangle$ is stored in the control register and the 1-qubit $|y\rangle$ in the target register. As the quantum circuit of Grover algorithm, an n-qubit control register is initialized to $|0\rangle^{\otimes n}$ and a 1-qubit target register to $|1\rangle$. In the next, n-qubit and 1-qubit Hadamard gates are applied on target and control registers sequentially. Moreover, the unitary function $U_f$ is applied on two registers and then, the sequence $H^{\otimes n}U_0^{\perp}H^{\otimes n}$ is applied on target register. Finally the output is measured and the result is evaluated. The classical post processing of the hybrid algorithm is run, if we achieve the desired result, otherwise Grover algorithm iterates the operator $G = H^{\otimes n}U_0^{\perp}H^{\otimes n}U_f$ that defined by the following sequence of transformations and named Grover Iterate. A feedback is needed to ensure the iteration these sequence transformations of G for $O(\sqrt{N})$ times.

The hybrid architecture of Shor's algorithm consists of two parts; first part is a reduction of the factoring problem to the problem of order-finding, which can be done on a classical computer. Second part is a quantum algorithm to solve the order-finding problem. We implement this algorithm based on hybrid architecture with 3 steps that steps 1 and 3 execute on classical computer and step 2 execute on quantum computer that here is simulated on classical computer. The complete cycle will be done as follows:

1. Initialize and run the classical part of the algorithm ( Pick a random number $a < N$, then compute GCD($a$, $N$) by using the Euclidean algorithm. If GCD($a$, $N$) $\neq 1$, then there is a nontrivial factor of $N$, so we are done. Otherwise, go to step 2 and use the period-finding algorithm to find $r$, the period of the function $f(x)=a^x \bmod N$, i.e. the smallest integer $r$ for which $f(x + r) = f(x)$ ).

2. Running the quantum algorithm by the quantum circuit ( Run the quantum order-finding algorithm):
   a. Initialize quantum circuit (Choose an integer $n$ so that $2^n \geq 2r^2$. The value $n = \lceil 2\log N \rceil$ will suffice).
   b. Prepare inputs state (Initialize an $n$-qubit $|0\rangle^{\otimes n}$ in control register, and an $n$-qubit $|\mathbf{1}\rangle = |00 \ldots 01\rangle$ in the target register).
   c. Execute the quantum portion of the algorithm (First, apply the QFT to the control register, and then apply unitary function c-$U_a^x$ on control and target registers, and finally apply the $QFT^{-1}$ to the control register).
   d. Measure the output of Machine State (Measure the control register to obtain an estimate $x_1/2^n$ of a random integer multiple of $1/r$).
   e. Evaluate Measurement (Use the continued fractions algorithm to obtain integers $c_1$ and $r_1$. Repeat step 2 to obtain another integer $x_2$ and a pair of integers $c_2$ and $r_2$, if no such pair of integers is found, output 'FAIL'.)
   f. Exit if desired result (Compute $r = \text{LCM}(r_1, r_2)$. If $a^r \bmod N = 1$, then output $r$, and go to step 3. Otherwise, output 'FAIL').
3. Run the second classical part of the algorithm( If $r$ is odd or $a^{r/2} \equiv -1 \bmod N$, go back to step 1 to repeat the algorithm, otherwise $\text{GCD}(a^{r/2} \pm 1, N)$ is a nontrivial factor of $N$. We are done).

The quantum period-finding method used to determine the order r of x modulo N. If r is even and $x^{r/2} \neq -1 \bmod N$, calculate $\text{GCD}(x^{r/2}-1, N)$ and $\text{GCD}(x^{r/2}-1, N)$. One of these should be a factor of N. If not, or if r is odd, repeat the algorithm, choosing a different x. The order of x modulo N is found by noting that we can calculate the modular exponentiation $x^a \bmod N$ for all a. We use two quantum registers, which will hold, respectively, a and $x^a \bmod N$. In the end, the control register measure to find the period of the function.

The creation of a machine that executes Shor's algorithm would have implications for security on the Internet, breaking the widely-used RSA public-key crypto system. The difficulty of cracking RSA is known to be related to the difficulty of factoring a large, composite number into its prime factors. Shor's algorithm can factor prime numbers readily on a quantum computer. However, Shor's algorithm demonstrates that a workable quantum computer can easily crack an RSA encryption scheme. All known algorithms for factoring an n-bit number on a classical computer take time proportional to $O(2^n)$ time and in the best known algorithm to $O(\exp(n^{1/3})$ time. But Shor's algorithm for factoring on a quantum computer takes time proportional to $O(n^3)$ (and with the optimized quantum circuit to $O(n^2 \log n)$).

## CONCLUSION

This study is proposed the hybrid architecture for the quantum algorithms. The quantum algorithms are as hybrid algorithms that consist of classical and quantum components. Moreover, the quantum portion of many algorithms is probabilistic; often need multiple runs to get the desired result. We designed and described relationship of classical and quantum part of algorithm that shows in Fig. 2. Also the following steps of each part of the hybrid algorithm are shown in Fig. 3. This flowchart will be used for design and implement the quantum algorithms. The implementation of the general plan on the quantum circuit of hybrid algorithms has been simulated on classical computer. This plan is useful for simulation and the architecture design of quantum computer. This plan is important to develop new quantum algorithms. A framework of quantum algorithm processing unit in the quantum computer is for future study.

## REFERENCE

1. Benioff, P., 1982. Quantum mechanical models of Turing machines that dissipate no energy. Phys. Rev. Lett., 48: 1581-1585. http://prola.aps.org/abstract/PRL/v48/i23/p1581_1
2. Deutsch, D., 1985. Quantum theory, the church-turing principle and the universal quantum computer. R. Soc. Lon. Ser., 400: 97-117. http://www.jstor.org/stable/2397601
3. McMahon, D., 2008. Quantum Computing Explained. Hoboken. John Wiley and Sons, Inc., New Jersey, ISBN: 978-0-470-09699-4, pp: 332.
4. Deutsch, D. and R. Jozsa, 1992. Rapid solution of problems by quantum computation. Proc. R. Soc. Lon., 439: 553-558. http://www.jstor.org/stable/52182
5. Simon, D., 1994. On the power of quantum computation. Proceeding of the 35th Annual Symposium on Foundations Computer Science, Nov. 20-22, IEEE Xplore Press, Santa Fe, NM., USA., pp: 116-123. DOI: 10.1109/SFCS.1994.365701

6. Shor, P.W., 1994. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev., 41: 303-332. http://portal.acm.org/citation.cfm?id=325514
7. Grover, L.K., 1996. A fast quantum mechanical algorithm for database search. Proceeding of the 28th Annual ACM Symposium on the Theory of Computation, May 22-24, ACM Press, New York, pp: 212-219. http://portal.acm.org/citation.cfm?id=237814.237866
8. Grover, L.K., 1997. Quantum Mechanics Helps in Searching for a Needle in a Haystack. Phys. Rev. Lett., 79: 325-328. http://prola.aps.org/abstract/PRL/v79/i2/p325_1
9. Kaye, P., R. Laflamme and M. Mosca, 2007. An Introduction to Quantum Computing. University Press, Oxford, ISBN: 0198570007, pp: 274.
10. Soltanaghaei, M.R., Z.A. Zukarnain, A. Mamat and H. Zainuddin, 2008. A quantum algorithm for minimal spanning tree. Proceeding of the 3rd International Symposium on Information Technology, Aug. 26-28, IEEE Xplore Press, Malaysia, pp: 1-6. DOI: 10.1109/ITSIM.2008.4632038
11. Soltanaghaei, M.R., Z.A. Zukarnain, A. Mamat and H. Zainuddin, 2008. A Hybrid Algorithm for the Shortest-Path Problem in the Graph. Proceeding of the International Conference on Advanced Computer Theory and Engineering, Dec. 20-22, IEEE Computer Society, Phuket Island, Thailand, pp: 251-255. DOI: 10.1109/ICACTE.2008.137
12. Soltanaghaei, M.R., Z.A. Zukarnain, A. Mamat and H. Zainuddin, 2009. A hybrid algorithm for finding shortest path in network routing. J. Theor. Applied Inform. Technol., 5: 360-364. http://www.jatit.org/volumes/research-papers/Vol5No3/14Vol5No3.pdf