Basic Quantum Algorithms and Applications

Marufa Rahmi
Lecturer, Dept. of CSE
Shah Jalal University Of Science and Technology

Debakar Shamanta
Lecturer, Dept. of CSE
Shah Jalal University Of Science and Technology

Ayesha Tasnim
Lecturer, Dept. of CSE
Shah Jalal University Of Science and Technology

ABSTRACT
Quantum computation, the ultimate goal of future computing, is an interesting field for researchers. The concept of quantum computation is based on basics of quantum mechanics. A quantum computer is a device for computation that makes direct use of quantum mechanical phenomena such as superposition and entanglement, to perform operations on data. The basic principle behind quantum computation is that quantum properties can be used to represent data and perform operations on these data. A quantum computer operates by manipulating the qubits with a fixed sequence of quantum logic gates. The sequence of gates to be applied is called a *quantum algorithm*. The field of quantum computation algorithm is fast moving and the scope is vast. Major quantum algorithms are summarized in this paper along with their applications.

General Terms
Quantum Algorithms.

Keywords
Qubit, Black box quantum computer known as an Oracle, Hadamard Transformation, Hadamard Gates, Superposition, Eigen value, Eigenstate.

1. INTRODUCTION
Researchers are working continuously with the quantum algorithms. Although the above four algorithms are still regarded as the primary algorithms, some modifications are done upon these algorithms. Though the total working in this field is still quite small, there are a number of algorithmic research areas where the quantum algorithms are applied and great advancements are achieved. There are also such fields where applying quantum algorithms will not always outperform the classical ones.

2. QUANTUM ALGORITHMS
Quantum algorithms are probabilistic algorithms [1]. The result obtained is not always correct but there is a high probability to get the correct solution. These algorithms are mainly based on implementing the quantum gates on qubits [2]. The elegance of these algorithms is due to quantum parallelism, interference and entanglement. All the transformations are used is unitary. They work on the qubits in a coherent system and the last stage is measurement which is known as decoherence to one of the possible outcomes. The basic algorithms are introduced here.

(1) Peter Shor's Factorizing Algorithm

(2) Lov Grover's Database Search Algorithm

(3) Simon's Algorithm for period finding.

(4) Deutsch-Jozsa Algorithm

3. PETER SHOR'S FACTORIZING ALGORITHM

3.1 Shor's Algorithm
This algorithm, first introduced by mathematician Peter Shor, is a quantum algorithm for integer factorization [3]. On a quantum computer, to factor an integer N, Shor's algorithm takes polynomial time in log$N$, specifically $O((\log N)^3)$. It demonstrates that integer factorization is in the complexity class **BQP**. This is exponentially faster than the best-known classical factoring algorithm, the general number field sieve and works in about $(e^{(logN)^{1/3}(loglogN^{2/3}})$. Peter Shor discovered the eponymous algorithm in 1994. It is very important because theoretically it can "break" the widely used public-key cryptography scheme known as RSA. RSA is based on the assumption that factoring large numbers is computationally infeasible for classical computers. Shor's algorithm shows that factoring is efficient on a quantum computer [1], [3].

3.2 Procedure of Shor's Algorithm
The problem statement is: given a composite number N, find an integer p, strictly between *1* and N, that divides N.

Shor's algorithm consists of two parts: A reduction of the factoring problem to the problem of order-finding, which can be done on a classical computer and a quantum algorithm to solve the order-finding problem.

3.2.1 Classical part of Shor's factorizing Algorithm

- Step 1- Pick a random number N, such that $a<N$
- Step 2- Compute gcd (a, N). This may be done using the Euclidean algorithm.
- Step 3- If gcd(a, N) $\neq$ 1, then there is a nontrivial factor of N, so it is done. Otherwise, use the period-finding subroutine to find r, the period of the following function: $(x) = a^x \bmod N$, the smallest positive integer r for which $f(x + r) = f(x)$.
- Step 4- If r is odd, go back to step 1.
- Step 5- If $a^{r/2} \equiv -1 \pmod N$, go back to step 1.
- Step 6- gcd($a^{r/2} \pm 1$, N) is a nontrivial factor of N. It is done.

3.2.2 Quantum part of Shor's factorizing Algorithm
Period-Finding Subroutine: The quantum circuits used for this algorithm are custom designed for each choice of N and the random a used in $f(x) = a^x \bmod N$. Given N, find $Q = 2^q$ such that $N^2 \leq Q < 2N^2$, which implies $Q / r > N$. The input and output qubit registers need to hold superpositions of values

from 0 to $Q-1$, and so have q qubits each. Using what might appear to be twice as many qubits as necessary guarantees that there are at least N different x which produce the same $f(x)$, even as the period r approaches $N/2$. Proceed as follows:

- Step1- Initialization of the registers to $Q^{-\frac{1}{2}}\sum_{x=0}^{Q-1}|x\rangle|0\rangle$Where x runs from 0 to $Q-1$. This initial state is a superposition of Q states.
- Step 2- Construction of $f(x)$ as a quantum function and applying it to the above state, to obtain $Q^{-\frac{1}{2}}\sum_x|x\rangle|f(x)\rangle$. This is still a superposition of Q states.
- Step 3- Applying the quantum Fourier Transform to the input register. This transform (operating on a superposition of power-of-two $Q = 2^q$ states) uses a Q^{th} root of unity such as $\omega = e^{2\pi i / Q}$ to distribute the amplitude of any given $|x\rangle$ state equally among all Q of the $|y\rangle$states, and to do so in a different way for each different x: $U_{QFT}|x\rangle = Q^{-\frac{1}{2}}\sum_y \omega^{xy}|y\rangle$. This leads to the final state $:Q^{-1}\sum_x\sum_y \omega^{xy}|y\rangle|f(x)\rangle$
  This is a superposition of many more than Q states, but many fewer than Q^2 states. Although there are Q^2 terms in the sum, the state$|y\rangle|f(x_0)\rangle$ can be factored out whenever x_0 and x produce the same value. Let $\omega = e^{2\pi i / Q}$ be a Q^{th} root of unity, r be the period of f, x_0 be the smallest of a set of x which yield the same given $f(x)$ (here $x_0<r$), and b run from 0 to $\lfloor(Q-x_0-1)/r\rfloor$ so that $x_0 + rb<Q$.

  Then ω^{ry} is a unit vector in the complex plane (ω is a root of unity and r and y are integers), and the coefficient of$Q^{-1}|y\rangle|f(x_0)\rangle$ in the final state is

$$\sum_{x:f(x)=f(x_0)} \omega^{xy} = \sum_b \omega^{(x_0+rb)y} = \omega^{x_0 y}\sum_b \omega^{rby}$$

Each term in this sum represents a *different path to the same result*, and quantum interference occurs constructive when the unit vectors ω^{ryb} point in nearly the same direction in the complex plane, which requires that ω^{ry} point along the positive real axis.

- Step 4- A measurement, some outcome y obtained in the input register and (x_0) in the output register. Since f is periodic, the probability of measuring some pair y and $f(x_0)$ is given by $|Q^{-1}\sum_{x:f(x)=f(x_0)}\omega^{xy}\big|^2 = Q^{-2}|\sum_b \omega^{(x_0+rb)y}\big|^2$.

Analysis now shows that this probability is higher, the closer unit vector ω^{ry} is to the positive real axis, or the closer yr/Q is to an integer.

Turn y/Q into an irreducible fraction, and extract the denominator r', which is a candidate for r. Check if $f(x) = f(x + r')$ $\Leftrightarrow a^r \equiv 1\ (mod\ N)$. If so, it is done. Otherwise, more candidates should be obtained for r by using values near y, or multiples of r'. If any candidate works, it is done. Otherwise, it should start from step 1 of the subroutine.

4. SIMON'S ALGORITHM

Simon's algorithm is one of the first quantum algorithms discovered which outperforms any known classical algorithm. Let$f: \{0,1\}^n \to \{0,1\}^n$ be such that for some$x \in \{0,1\}^n$ it is true for all,$z \in \{0,1\}^n$,$f(y) = f(z)$ if and only if $y = z$ or$y \oplus z = x$.

This algorithm provides an exponential improvement in time over any known classical algorithm. To find x using a classical randomized algorithm $\Omega(2^{n/2})$ queries of f would be required. Using Simon's algorithm it is possible to find a solution with high probability using $O(n)$ queries of f. It was also the inspiration for Shor's algorithm.

4.1 Procedure of the Algorithm

The problem can be stated as a decision problem which goal is to decide whether or not there is a period that is whether f is 2 to 1 or 1 to 1. *Simon's problem* is an instance of an oracle problem which is classically hard, even for probabilistic algorithms, but tractable for quantum computers [4].

Classically the problem is hard because the probability to find two identical elements x and y after 2^(N/4) queries is less than 2^(−N/2). Simon's quantum solution is as the following [4]:

- Start with a state vector $\langle H|0\rangle)^{\otimes N}|0\rangle^{\otimes N}$
- Run the oracle once to make the state vector $2^{-N/2}\sum_x|x\rangle|f(x)\rangle$
- Measure the second register; if the measurement outcome is $f(x_0)$, then the state vector of the first register will be $\frac{1}{\sqrt{2}}(|x_0\rangle + x_0 \oplus p\rangle)$
- Applying a Hadamard gate to each of the N remaining qubits leads to

$$\frac{1}{2^{(N+1)/2}}\sum_y((-1)x_0.y + (-1)^{(x_0\oplus p).y})|y\rangle$$

$$= \frac{1}{2^{(N-1)/2}}\sum_{p.y-0}(-1)^{x_0.y}\,|y\rangle$$

Final measurement of the first register in computational basis, will give a value y which is such that y.p= 0 modulo 2.

Repeating this procedure in order to get N − 1 linearly independent vectors $y_1,\ldots,y_{N-1}$ p can be determined from the set of equations $\{y_i.\ p = 0\}$. To this end there should be a procedure to query the oracle O(N) times.

Hence an exponential speed up is obtained compared to any classical algorithm. Simon's algorithm has much common characteristics with Shor's algorithm; both look for the period of a function [4], yield an exponential speed-up and both make use of classical algorithms in a post processing step. This algorithm was the inspiration of Shor's work.

5. GROVER'S DATABASE SEARCH ALGORITHM

5.1 Grover's Algorithm

This is a quantum algorithm for searching an unsorted database with *N* entries in $O(N^{1/2})$ time and using $O(\log N)$ storage space [5]. It was invented by Lov Grover in 1996.

Classically, searching an unsorted database requires a linear search, which is *O(N)* in time. Grover's algorithm, which takes $O(N^{1/2})$ time, is the fastest possible quantum algorithm for searching an unsorted database [6]. It provides "only" a quadratic speedup, compared to other quantum algorithms, which may provide exponential speedup over their classical counterparts. The quadratic speedup is considerable when *N* is large.

Like many other quantum algorithms, Grover's algorithm is probabilistic because it gives the correct answer with high probability. The probability of failure can be decreased by repeating the algorithm [6].

Grover's algorithm can be described as "inverting a function" [5]. If there is a function *y=f(x)* that can be evaluated on a quantum computer, this algorithm allows us to calculate *x* when given *y*. Inverting a function is related to the searching of a database in the sense there can be a function that produces a particular value of *y* if *x* matches a desired entry in a database, and another value of *y* for other values of *x*. Grover's algorithm can also be used for estimating the mean and median of a set of numbers, and for solving the Collision problem. It can also be used to solve NP-complete problems by performing exhaustive searches over the set of possible solutions [7].

5.2 Procedure of Grover's Algorithm

Let us consider an unsorted database with N entries. The algorithm requires an *N*-dimensional state space *H*, which can be supplied by $\log_2 N$ qubits.

Let us number the database entries by 1, 2... *N*. Choose an observable, *Ω*, acting on *H*, with *N* distinct eigenvalues whose values are all known. Each of the eigenstates of *Ω* encodes one of the entries in the database, in a described manner. Eigenstates are denoted as $\{|1\rangle, |2\rangle, \dots, |N\rangle$ (using bra-ket notation) and the corresponding eigenvalues by$\{\lambda_1, \lambda_2, \dots, \lambda_N\}$.

A unitary operator is provided, U_ω, which acts as a subroutine that compares database entries according to some search criterion. The algorithm does not specify how this subroutine works, but it must be a *quantum* subroutine that works with superpositions of states. Furthermore, it must act especially on one of the eigenstates, $|\omega\rangle$, which corresponds to the database entry matching the search criterion. To be precise, it is required U_ω to have the following effects: $U_\omega|\omega\rangle = -|\omega\rangle, U_\omega|x\rangle = |x\rangle\ ,\ for\ all\ x \neq \omega$

Our goal is to identify this eigenstate $|\omega\rangle$, or equivalently the eigenvalue ω, that U_ω acts especially upon. Two unitary operators are defined as follows: $U_\omega = I - 2|\omega\rangle\langle\omega|$ and $U_s = 2|s\rangle\langle s| - I$after application of the two operators ($U\omega$ and U_s), the amplitude of the searched-for element increases.

And this is one Grover iteration *r*. $N=2^n$, *n* is number of qubits in blank (zero) state.

$U_\omega|s\rangle = |s\rangle - \frac{2}{\sqrt{N}}|\omega\rangle$ And $U_s\left(|s\rangle - \frac{2}{\sqrt{N}}|\omega\rangle\right) = \frac{N-4}{N}|s\rangle + \frac{2}{\sqrt{N}}|\omega\rangle$

The steps of Grover's algorithm are as follows:

1. Initialize the system to the state$|s\rangle = \frac{1}{\sqrt{N}}\sum_{x=1}^{N}|x\rangle$.
2. Perform the following "Grover iteration" *r(N)* times. The function *r(N)* is described below.
    1. Apply the operator $U_\omega = I - 2|\omega\rangle\langle\omega|$.
    2. Apply the operator$U_s = 2|s\rangle\langle s| - I$.
3. Perform the measurement Ω. The measurement result will be λ_ω with probability approaching 1 for N>>1. From λ_ω, ω may be obtained.

5.3 Generalization of Grover's Algorithm for Multiple objects

There is a generalization of Grover's search algorithm for the data where the number of objects satisfying the search criterion is greater than 1 [8].

Let a database$\{\omega_i | i = 1, 2, \dots, N\}$, with corresponding orthonormal eigenstates $\{|\omega_i\rangle : i = 1, 2 \dots N\}$. Let *f* be an oracle function such that

$$f(\omega_j) = \begin{cases} 1, & j = 1,2, \dots, \ell \\ 0, & j = \ell+1, \ell+2, \dots, N \end{cases}$$

Here the ℓ elements$\{\omega_j | 1 \le j \le \ell\}$ are the desired objects of search. For simplicity it is assumed that the searched for elements are the first ℓ items in the list while in reality it would be random. Let H be the Hilbert space generated by the orthonormal basis of the discussed database, the linear operation in terms of the oracle function *f* as follows:

$I_L|\omega_j\rangle = (-1)^{f(\omega)}|\omega_j\rangle, j = 1,2, \dots N$, Which is equivalent to $I_L = \boldsymbol{I} - 2\sum_{j=1}^{\ell}|\omega_j\rangle\langle\omega_j|$,since $I_L$is linear.

The overall states are defined as$|s\rangle$ where$|s\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}|\omega_i\rangle$and the searched for elements are the first ℓ elements. Another operation is defined as $I_s = \boldsymbol{I} - \boldsymbol{2}|s\rangle\langle s|$.

This operation is unitary and hence quantum-mechanically admissible. This is explicitly known and constructible with the Walsh–Hadamard transformation.

The generalized Grover Search Mechanism for multiple objects searching is constructed as$U = -I_s I_L$.

The total number of iterations needed depends on the number ℓ. The complexity of original Grover's Search is $O(N^{1/2})$and the presence of multiple object satisfying the search criteria speeds up the algorithm and it becomes $O((N/L)^{1/2})$ [6], [8].

6. DEUTSCH'S ALGORITHM

6.1 Original Deutsch's Algorithm

The Deutsch algorithm is an elementary quantum algorithm which is proposed by David Deutsch [9]. Although it is of little practical use, it is one of the first examples of a quantum algorithm that is more efficient than any possible classical algorithm. It uses the power of quantum computing such as quantum parallelism, interference and entanglement.

6.2 Procedure of Deutsch's Algorithm

In the Deutsch problem, A black box quantum computer known as an oracle is given, that implements the function $f: \{0,1\} \rightarrow \{0,1\}$.

Now the condition $f(0) = f(1)$ needs to be checked. It is equivalent to check $f(0) \oplus f(1)$(where $\oplus$ is addition modulo 2), if zero, then *f* is constant, otherwise *f* is not constant. It is not concerned to find the value or outcome of *f(x)* itself.

To find the answer classically, one needs to query for both $x = 0$ and $x = 1$, hence two queries are required [8]. Quantum mechanically this can be solved in just one query. The figure represents the circuit for Deutsch's Algorithm.

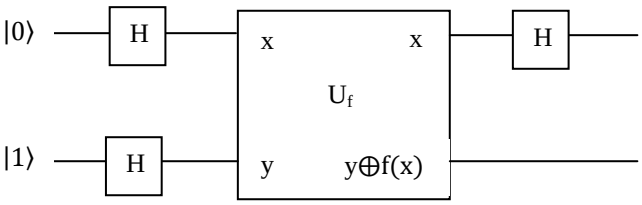

Fig: Circuit diagram of Deutsch's algorithm

Here, given a function $f: \{0,1\} \rightarrow \{0,1\}$, two qubits $|x, y\rangle$ are used and transferred them into $|x, y \oplus f(x)\rangle$. Two qubits are used to preserve reversibility, to keep the value of input x after the oracle performs. The second qubit *y* acts as a result register. Let *Uf* be the unitary transform that implements the function and maps $|x\rangle|y\rangle$ to $|x\rangle|f(x) \oplus y\rangle$.

The procedure begins with the two qubits in the state $|0\rangle|1\rangle$ and then a Hadamard transform applied to each qubit. This yield $\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$.

After applying the function to the current state:

$$\frac{1}{2}(|0\rangle(|f(0) \oplus 0\rangle - |f(0) \oplus 1\rangle) + |1\rangle(|f(1) \oplus 0\rangle - |f(1) \oplus 1\rangle))$$

$$= \frac{1}{2}((-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle))$$

$$= (-1)^{f(0)}\frac{1}{2}\left(|0\rangle + (-1)^{f(0) \oplus f(1)}|1\rangle\right)(|0\rangle - |1\rangle).$$

The last bit is ignored and the global phase and therefore have the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{f(0) \oplus f(1)}|1\rangle\right).$$

Applying a Hadamard transform to this state:

$$\frac{1}{2}(|0\rangle + |1\rangle + (-1)^{f(0) \oplus f(1)}|0\rangle - (-1)^{f(0) \oplus f(1)}|1\rangle)$$

$$= \frac{1}{2}((1 + (-1)^{f(0) \oplus f(1)})|0\rangle + (1 - (-1)^{f(0) \oplus f(1)}|1\rangle).$$

If the result of measurement is a Zero, $f(0) \oplus f(1) = 0$. Therefore the function is constant and otherwise it is balanced.

Here U_f is applied to 0 and 1 simultaneously. This is known as quantum parallelism.

The solution is to use another quantum mechanical property named *interference*.

Deutsch's algorithm like all known quantum algorithms which provide exponential speedup over classical systems, answers a question about a global property of a solution space [10]. These are often called *promise problems*, where the structure of the solution space is promised to be of some form and by carefully using superposition, entanglement and interference the information about that structure can be extracted. Exponential improvement is possible for quantum parallelism. Quantum computers can only provide square-root improvement to the query-based problems [6].

7. DEUTSCH-JOZSA ALGORITHM

The **Deutsch-Jozsa algorithm** is a quantum algorithm, proposed by David Deutsch and Richard Jozsa in 1992 with improvements by R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca in 1998. This algorithm has a little practical use but it is one of the first examples of a quantum algorithm that is more efficient than any possible classical algorithm [11], [12].

7.1 The Algorithm

In the Deutsch-Jozsa problem, A black box quantum computer known as an oracle is given that implements the function, $f: \{0,1\}^n \rightarrow \{0,1\}$. It is said that the function is either constant (0 on all inputs or 1 on all inputs) or *balanced* (returns 1 for half of the input domain and 0 for the other half); the task then is to determine if *f*is constant or balanced by using the oracle.

7.2 Procedure of Deutsch-Jozsa Algorithm

For a conventional deterministic algorithm, $2^{n-1} + 1$ evaluations of *f* will be required in the worst case (and in best case need only 2 queries, if the function is balanced), where *n* is number of bits/qubits. For a conventional randomized algorithm, constant *k* evaluations of the function are enough to produce the correct answer with a high probability. If a correct answer is wanted always, $k = 2^{n-1} + 1$ evaluations are required. The Deutsch-Jozsa quantum algorithm produces an answer that is always correct with a single evaluation of *f*.

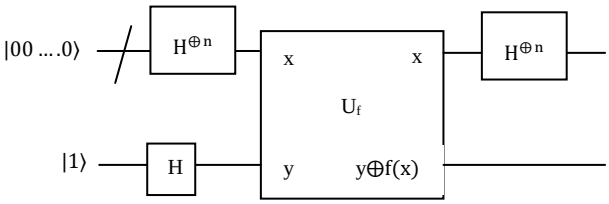

Fig: Classical solution of Oracle

The Deutsch-Jozsa Algorithm is the multi qubit generalization of the Deutsch's Algorithm. The procedure begins with the n+1 bit state$|0\rangle^{\otimes n}|1\rangle$. The first n bits are each in the state$|0\rangle$ and the final bit is$|1\rangle$. After applying a Hadamard transformation to each bit, the state is then,

$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\,(|0\rangle-|1\rangle)$.

The function *f* is implemented as quantum oracle which maps the state $|x\rangle|y\rangle$to $|x\rangle|y\oplus f(x)\rangle$. After applying that, the state is,

$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\,(|f(x)\rangle-|1\oplus f(x)\rangle)$.

The two possibilities for *f*(*x*), that is$f(0)=f(1)$ and$f(0)\neq f(1)$ reduces to

$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle\,(|0\rangle-|1\rangle)$.

Ignoring the last qubit and applying a Hadamard transformation to each bit, the probability of measuring$|0\rangle^{\otimes n}$,$\left|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\right|^2$ which evaluates to 1 if *f*(*x*) is constant and 0 if *f*(*x*) is balanced.

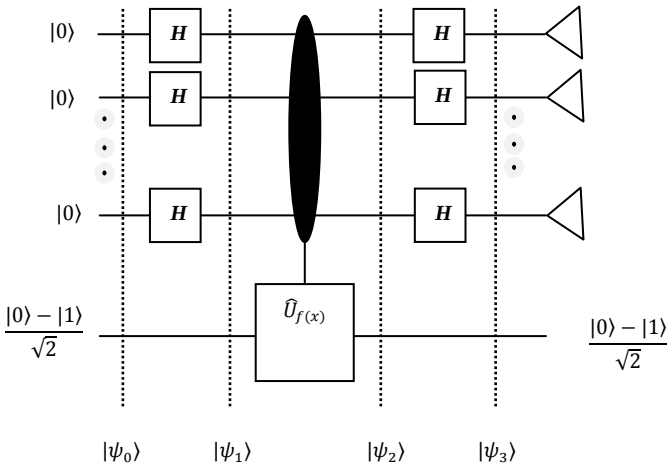

Figure: Circuit for the Deutsch-Jozsa Algorithm

8. NOTABLE APPLICATIONS OF QUANTUM ALGORITHMS

Researchers are working continuously with the quantum algorithms. Although the above four algorithms are still regarded as the primary algorithms, some modifications are done upon these algorithms. Though the total working in this field is still quite small, there are a number of algorithmic research areas where the quantum algorithms are applied and great advancements are achieved. There are also such fields where applying quantum algorithms will not always outperform the classical ones. Below some applications of quantum algorithms which are done in recent years are noted.

8.1 Primality Test with Quantum Factorization

Some researchers suggest a probabilistic quantum implementation for a specific Primality test method using Shor's algorithm. There O ($\log^3$ N log log N log loglog N) elementary q-bit operations are required to determine the primality of a number N. This can be thought of as (asymptotically) the fastest known Primality test. The potential power of quantum mechanical computers is once again revealed by this way [3], [13].

8.2 Factoring Number and Finding Divisors

Peter Shor's Algorithm is generalized to find the prime factors of an integer. Special quantum circuit design is proposed to find the divisors of a number. Wiring diagrams are given for a quantum algorithm processor in CMOS to compute, in parallel, all divisors of an n-bit integer. The lines required in a wiring diagram are proportional to n and the execution time is proportional to the square of n [3], [8], [13].

8.3 Sorting by Quantum Algorithms: Time-Space trade off

In general a quantum algorithm based on only comparisons outperforms the classical sorting algorithms by only a constant factor in time complexity. A quantum sort is any sorting algorithm that runs on a quantum computer. Any comparison-based quantum sorting algorithm would take at least Ω(*nlogn*) steps, which is already achievable by classical algorithms. It is different in a space bounded setting. For all storage bounds n/log n ≥ S ≥ log3n, one can devise a quantum algorithm that sorts n numbers (comparisons comparison based) in time T=O(n3/2 log3/2 n/√S) [14].

8.4 Quantum Searching

Grover's algorithm offers searching in the unsorted database in less complexity. Any quantum algorithm searching an ordered list of n elements needs to examine at least log n/12-O(1) of them [14][15]. Classically, log n queries are both necessary and sufficient. This reveals that quantum algorithms can achieve only a constant speedup for this problem.

8.5 Application in NP-Complete Problems

NP complete problems need an exhaustive search over the total range of possibilities. Quantum search is used to perform the search operation faster. Solutions of some np complete problems such as traveling salesman problem, finding all solution of n-queen, Hamiltonian path/cycle finding etc are the field of ongoing research.

The Hamiltonian cycle problem is to determine whether a given graph has a Hamiltonian cycle or not. Some researchers used undirected graphs with varied number of vertices and showed how to determine the existence of a Hamiltonian cycle in a given graph [7]. They illustrated how quantum search can be applied to obtain the solution of the Hamilton cycle problem much faster than the classical approach.

8.6 Application in Graph Theories

Graph theory is the vast field where quantum approach can be applied to outperform the classical algorithms [7]. By using the quantum search as subroutines in the existing algorithms for finding shortest path, minimum spanning tree, least weighted cycle etc, it is found to perform better.

8.7 Other Works

Between other works finding modal value of data, generating random number, finding the solution of Pell's equation, pattern matching, different type of satisfiability (SAT) problems etc are noteworthy[16].

9. CONCLUSION

Quantum algorithms are a field of growing interest within the theoretical computer science and the physics community. A lot of research works are going on this field. Within a few decades our classical computing is going to be replaced by faster (possibly) quantum counterparts. The field of quantum computing algorithms are very fast moving and coming up with innovative features. We are feeling elevated for getting mixed with it.

10. ACKNOWLEDGMENTS

We want to extend our heartfelt thanks to some persons for their cordial help. The person who always helps us greatly is, Dr. Susanta Kumer Das, Professor of Physics, Shah Jalal University of Science and Technology. He relentlessly gave us enthusiasm for the research and gave full support for all kinds of resources we needed. And we also want thank the scientists and researchers who developed the algorithms and analyze these.

11. REFERENCES

[1] VlatkoVedral, Martin B. Plenio, Basics of Quantum Computation, Progress in quantum electronics, vol 22, (1998).

[2] John Preskill, Making Weirdness Work: Quantum Information and Computation.

[3] Peter W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.

[4] J. Eisert, M.M. Wolf, Quantum Computing.

[5] Grover L.K, A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212.

[6] Lov K. Grover (1996). "A fast quantum mechanical algorithm for database search", Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing:212–219.

[7] Vidya Raj C and M. S. Shivakumar, Applying Quantum Algorithm to Speed Up the Solution of Hamiltonian Cycle Problems, IFIP International Federation for Information Processing, Springer Boston, Volume 228/2007:53-61

[8] Goong Chen, Stephen A. Fulling, Jeesen Chen, Generalization of Grover's Algorithm to Multiobject Search in Quantum Computing, Part I: Continuous Time and Discrete Time, quant-ph/0007123, Jul 2000.

[9] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[10] Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[11] David Deutsch and Richard Jozsa (1992)."Rapid solutions of problems by quantum computation". Proceedings of the Royal Society of London A439: 553.

[12] Cleve, A. Ekert, C. Macchiavello, and M. Mosca (1998). "Quantum algorithms revisited", Proceedings of the Royal Society of London A454: 339–354.

[13] Hartmut Klauck, Quantum time-space tradeoffs for sorting, Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, San Diego, CA, USA, Session 2A : 69 – 76(2003).

[14] P. Høyer, J. Neerbek, Y. Shi (2001). Quantum complexities of ordered searching, sorting, and element distinctness, 28th International Colloquium on Automata, Languages, and Programming: 62-73.

[15] Andris Ambainis , A Better Lower Bound for Quantum Algorithms Searching an Ordered List, Proceedings of the 40th Annual Symposium on Foundations of Computer Science, Page: 352 (1999).

[16] Sean Hallgren, Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem, Proceedings of the 34th ACM Symposium on Theory of Computing, 2002 Pages: 653-658.