QUANTUM ALGORITHMS AND HARD PROBLEMS

Vidya Raj C.
Dept of CS & Engg.
NIE, Mysore – 570 008
Karnataka, India
vidya_rajc@yahoo.com

Phaneendra H. D.
Dept of CS & Engg.
NIE, Mysore- 570008
Karnataka, India
hdphanee@yahoo.com

Dr. Shivakumar M.S.
Principal
NIE, Mysore-570008
Karnataka, India
skumar@vtu.ac.in

Abstract

Any computation is necessarily a physical process. The current drive towards increasing speed and miniaturization of computers lead modern technology towards the subatomic domain - Quantum computing, where strange quantum behavior takes over from familiar classical notions. The quantum computers have the ability to solve problems with varied computational complexities and more importantly hard problems. A problem is said to be hard, if the best possible algorithm requires exponential resources. This exponential quantum parallelism *is the basis for the quantum speed-up of many algorithms, and has made the construction of fast algorithms for quantum computers possible. Various proposals exist for modeling and devising procedures on computational problems related to factoring, searching, quantum counting, generalization of quantum algorithms etc.[1]. In this paper, we have discussed the importance of algorithms such as order-finding, factoring and quantum searching which require exorbitant resources for their solution which otherwise is impossible on a classical computer*

Keywords: *Quantum computer; quantum algorithms; hard problem; factoring; qubits.*

1. INTRODUCTION

As our technology rushes forward, several factors work together to push us toward the quantum computing world, and push out the classical silicon-based chips. These factors include scaling in size, energy consumption, and economics of building leading-edge computers, and new applications that are available with quantum computers that cannot be run on the classical computers.

Quantum computing is an important field of research that applies concepts of quantum physics to building more efficient computers. Although only rudimentary quantum computers have been built so far, many researchers believe that quantum computing has great potential and the quantum computers can efficiently perform some tasks such as factoring which are otherwise not feasible on a classical computer.

One of the most useful ways of solving a problem in mathematics or computer science is to transform it into some other problem for which a solution is known. An important discovery of quantum computation has been that some such transformations can be computed much faster on a quantum computer than on a classical computer, a discovery which has made the construction of fast algorithms for quantum computers possible. And one such transformation is the quantum Fourier transforms [9].

1.1 Quantum Computation

Quantum Computation is the field of study which focuses on developing computer technology based on the principles of quantum theory.

The basic variable used in quantum computing is a qubit, represented as a vector in a two dimensional complex Hilbert space where $|0>$ and $|1>$ form a basis in the space. The difference between qubits and bits is that a qubit can be in a state other than $|0>$ or $|1>$ whereas a bit has only one state, either 0 or 1. It is also possible to form linear combination of state, often called superposition[9]. The state of a qubit can be described by

$$|\Psi> = \alpha|0> + \beta|1>$$

Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI'06)
Y.Y. Yao, Z.Z. Shi, Y. Wang, and W. Kinsner (Eds.)
1-4244-0475-4/06/$20.00 ©2006 IEEE

The numbers α and β are complex numbers. The special states $|0>$ and $|1>$ are known as computational basis states. We can examine a bit to determine whether it is in the state 0 or 1 but we cannot directly examine a qubit to determine its quantum state, that is values of α and β. When we measure a qubit we get either the result 0, with probability $|\alpha|^2$ or the result 1, with probability $|\beta|^2$, where $|\alpha|^2 + |\beta|^2 = 1$, since the probabilities must sum to one.

Consider the case of two qubits. In two classical bits there would be four possible states, 00, 01, 10 and 11. Correspondingly, a two qubit system has four computational basis states denoted $|00>$, $|01>$, $|10>$ and $|11>$. A pair of qubits can also exist in a superposition of these four states, so the quantum state of two qubits involves associating a complex coefficient, sometimes called amplitude, with each computational basis state, which is given as

$$|\Psi> = \alpha_{00}|00> + \alpha_{01}|01> + \alpha_{10}|10> + \alpha_{11}|11>$$

The logic that can be implemented with qubits is quite distinct from Boolean logic, and this is what has made quantum computing exciting by opening new possibilities [9].

1.2 Quantum Algorithms

Quantum computers are devices that use quantum mechanical phenomenon such as superposition and entanglement, to perform operations on data. The data are measured by qubits instead of bits in a classical computer.

The hardness of factorization over the classical computers has led to the development of more powerful algorithms based on the principles of quantum mechanics, called quantum algorithms.

There are two known algorithms, which would allow quantum computers to threaten cryptographic systems that are in widespread use today. They are Shor's algorithm and Grover's algorithm. Shor's algorithm allows for factoring large numbers on a quantum computer in polynomial time [2]. Grover's algorithm allows for searching an unsorted database with quadratic speed [8].

2. PROBLEMS ON QUANTUM COMPUTERS

Even though quantum algorithms cannot efficiently solve NP-Complete problems, there are several interesting problems that apparently do not have efficient solutions in classical computing but do not appear to be NP-complete either. And these are the good candidates [1][9] for efficient quantum algorithms:

- Order-finding
- Factoring
- Discrete logarithm
- Searching of unsorted database
- Analysis of large amounts of data
- Lattice problems
- Graph isomorphism
- Group problems
- Simulating Physics

2.1 Order - finding

It is considered to be a "hard" problem on a classical computer. It is used for determining the order for some specified x and N where, x and N are positive integers, x < N, with no common factors, the order of x modulo N is defined to be the least positive integer r such that $x^r = 1 (\text{mod } N)$. Problems like order-finding can be solved using quantum Fourier transforms. This algorithm becomes important for factorization problems. This can be illustrated as shown below.

Table 1. Order-finding algorithm, find min r > 0 such that x^r = 1 (mod N)."

	x = 5	N = 21
r	x^r	x^r mod N
0	1	1
1	5	5
2	2	4
3	12	2
4	62	1
5	312	1
6	1562	1
7	7812	5
8	39062	4
9	195312	2
10	976562	1
11	4882812	1

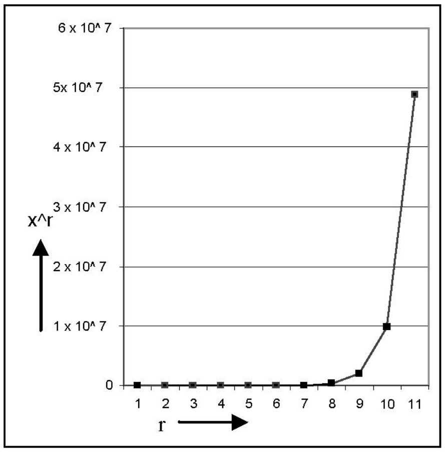

Figure 1. Order-finding algorithm, r in x-axis Vs x^r in y-axis."

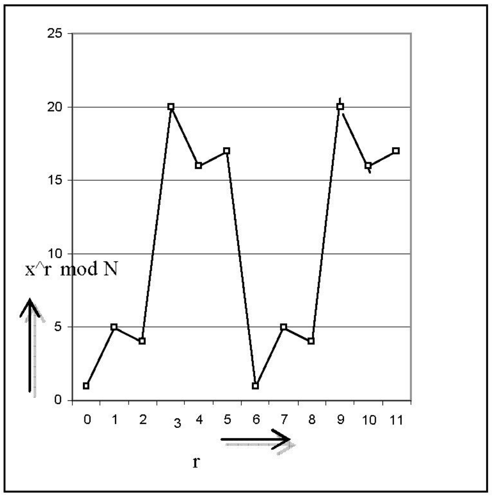

Figure 2. Order-finding algorithm, r in x-axis Vs. x^r mod N in y-axis."

2.2 Factoring

Is a method of finding the prime factors for a given number. Factoring is generally considered to be a "hard" problem, especially for numbers that are products of 2 large primes. An efficient algorithm for factoring large integers, known as Peter Shor's algorithm[1] aroused a great deal of interest among scientists and the lay public and also heralded the birth of the new field of quantum computation[10][4].

The algorithm has two parts. The first part of the algorithm turns the factoring problem into the problem of finding the period of a function, and it may be implemented classically. The second part of the algorithm finds the period using the quantum fourier transforms, and is responsible for the quantum speedup.

Some important features of Peter Shor's algorithm: Shor's algorithms are polynomial time algorithms for factoring integers and computing discrete logarithms. Shor's algorithm is probabilistic: it gives the correct answer with high probability, and the probability of failure can be decreased by repeating the algorithm. Shor's quantum algorithm factors a number N in O((log N)^3) time and O(log N) space. Using Shor's algorithm it was possible for a quantum computer with 7 qubits, to factorize number 15 into 3 and 5 [5][9]. The number 15 was factored using 10^{18} identical molecules, each containing 7 atoms. Shor's algorithm can crack RSA in polynomial time and so also other crpto systems.

"Table 2. Performance of factoring a 300 digit number on classical computer and quantum computer [3]."

	Best classical algorithm	Shor's quantum algorithm
Operations	10^{24} steps	10^{10} steps
Speed (THz)	150,000 years	< 1 second

"Table 3. Performance comparison of Shor's factorization algorithm with other factorization algorithms[3][7]."

Number of digits (public key size)	Number of bits (approx.)	Time-line	Effort level (MIPS - Years)	Algorithm
100	332	1991	7	Quadratic sieve
110	365	1992	75	Quadratic sieve
120	398	1993	830	Quadratic sieve
129	428	1994	5000	Quadratic sieve
130	431	1996	1000	Generalized number field sieve
140	465	1999	2000	Generalized number field sieve
155	512	1999	8000	Generalized number field sieve
300	1024	Yet to crack	10^11	Generalized number field sieve
300	1024	Yet to crack	10^7	Special number field sieve
300	1024	1994	< 1 second	Shor's quantum factorization

2.3 Searching of Unsorted Database

Grover's algorithm 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. 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. It provides "only" a quadratic speedup, unlike other quantum algorithms, which can provide exponential speedup over their classical counterparts. However, even quadratic speedup is considerable when N is large. Like many quantum computer algorithms, Grover's algorithm is probabilistic, in the sense that it gives the correct answer with high probability. The probability of failure can be decreased by repeating the algorithm

“Table 4: Performance comparison of searching an unsorted database on a classical computer and quantum computer [6][8].”

Algorithm	Linear search	Binary search	Quantum search
Uses	Bits	Bits	Qubits
Operations in time	$O(N)$	$O(\log N)$ *to base 2*	$O(N^{1/2})$

2.4 Lattice Problems & Graph Isomorphism

Answers **“**What’s Next?”. Lattice problems are the problem of finding a short vector in a lattice. Given a lattice in d dimensions, is it possible to efficiently find a vector that is not much longer than the shortest vector in this lattice? This becomes hard for large d [11]. These problems have applications to cryptography**.** Graph Isomorphism is an example of a more general hidden subgroup problem. Given two graphs, describing a relational structure, it finds if there is a permutation of the nodes which renders them identical?

3. CONCLUSIONS

The design of faster-than-classical quantum algorithms for important algorithmic problems has been an interesting intellectual adventure and achievement. Their existence keeps being one of the key stimuli to those trying to overcome enormous technology problems for building quantum computers. Still, quantum computers may never be general-purpose computing devices and are more likely to be targeted at massive number-crunching problems like encryption and decryption, searches of huge databases and simulations of quantum physical states. And of course, in years to come quantum computers will make the computation a property of matter and matter software. The purpose of this paper is to indicate how we were able to explore some of the wonderful contributions of fast quantum algorithms which are attempting to solve problems which are intractable, not feasible on a classical system.

Acknowledgements

The authors would like to thank all the authors and publishers for giving us enough information on this emerging area of computing.

Our thanks are due to our institute, “The National Institute of Engineering, Mysore” for all the encouragement and financial support through the Technical Education Quality Improvement Programme of the country.

References

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring,” *Proceedings 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, USA, IEEE Computer Society Press, pp.124-134, 1994.

[2] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM J. Computing* 26, pp.1484-1509, 1997.

[3] Stephen Barlett, “Lecture on quantum computing,” *NITP Summer School*, Adelaide, Australia, 2003.

[4] Feynman, R. P. "Simulating Physics with Computers," *International Journal of Theoretical Physics*, vol. 21,pp. 467-488, 1982.

[5] IBM's announcement of the first actual execution of the algorithm, in an issue of “Nature”. published in the December 19, 2001.

[6] Apoorva Patel, “Quantum Database Search can do without Sorting,” *quant-ph/0012149.*

[7] William Stallings, “Cryptography and Network Security Principles and Practices ”, Pearson education, Third Edition.

[8] Grover L.K,”A Fast Quantum mechanical Algorithm for Database Search,” In proceedings of the *28th Annual ACM Symposium on the Theory of Computing*, pp. 212-219, *quant-ph/9605043*, 1996.

[9] M. A. Nielsen and I. L.Chuang, "Quantum Computation and Quantum Information,” Cambridge University Press, 2002.

[10] Deutsch, D. "Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer," Proc. Roy. Soc. Lond. A400, 1985.

[11] Peter W. Shor, “Progress in quantum algorithms,” Sept 2005.