International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:6, No:3, 2012

Application of Genetic Algorithms for Evolution of Quantum Equivalents of Boolean Circuits

Swanti Satsangi, Ashish Gulati, Prem Kumar Kalra and C. Patvardhan

Abstract**—**Due to the non- intuitive nature of Quantum algorithms, it becomes difficult for a classically trained person to efficiently construct new ones. So rather than designing new algorithms manually, lately, Genetic algorithms (GA) are being implemented for this purpose. GA is a technique to automatically solve a problem using principles of Darwinian evolution. This has been implemented to explore the possibility of evolving an n-qubit circuit when the circuit matrix has been provided using a set of single, two and three qubit gates. Using a variable length population and universal stochastic selection procedure, a number of possible solution circuits, with different number of gates can be obtained for the same input matrix during different runs of GA. The given algorithm has also been successfully implemented to obtain two and three qubit Boolean circuits using Quantum gates. The results demonstrate the effectiveness of the GA procedure even when the search spaces are large.

Keywords**—**Ancillas, Boolean functions, Genetic algorithm, Oracles, Quantum circuits, Scratch bit

I. Introduction

THERE has been considerable progress in the field of classical algorithm design over the last few decades with some of the most outstanding people working in the area. Good algorithms have been designed for many problems. Where the problems have turned out to be hard, good approximation algorithms have been designed. Even if this is also not successful for some problem, good "meta-heuristics" are available and the computational power of modern computers can be put to good use using these techniques to at least an acceptable or satisfying solution if not a probably best one. On the other hand, there exist only a handful of quantum algorithms that are more efficient than their classical counterparts; such algorithms were invented by Shor in 1994 and then Grover in 1996. A lack of invention since Grover's algorithm has been commonly attributed to the non-intuitive nature of quantum algorithms to the classically trained person. Moreover, any quantum algorithm would be acceptable only if it is significantly "better" than the existing ones for the problem at hand [3]. Thus, the idea of using computers to automatically generate quantum algorithms, based on an evolutionary model has emerged. This gives the motivation to use Genetic algorithms to evolve quantum algorithms, with the hope that their power to search vast, complex and unknown spaces can discover new and superior to existing quantum algorithms.

Swanti Satsangi is a research scholar in the Department of Physics and Computer Science at Dayalbagh Educational Institute, Agra and the Indian Institute of Technology, Delhi, India. She is also currently working as a Project Associate in the Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, India. (email: swantis@gmail.com)

Ashish Gulati is an undergraduate student in Division of Electronics and Communications Engineering, Netaji Subhas Institute of Technology, Delhi, India. (email: ashishgulati11@gmail.com)

Prem K. Kalra is a Professor in the Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, India. (email: pkalra@cse.iitd.ac.in)

C. Patvardhan is a Professor in the Department of Electrical Engineering, Dayalbagh Educational Institute, Agra, India. (email: cpatvardhan@gmail.com)

II. Quantum versus Classical Circuits

Just like classical circuits are composed of different gates like AND, OR, NOT etc. Quantum circuits are composed of quantum gates like Hadamard, CNOT, Toffoli etc. In quantum computation the qubit is the basic unit of information. In Bra-Ket notation, a qubit is a normalized vector in a two dimensional Hilbert space

$$|\psi> = a|0> + b|1>$$

such that

$$a^2 + b^2 = 1$$

and $|0>$ and $|1>$ are the basis states. The quantum system is described by a superposition of the basis states whereas a classical binary system can only settle in one of the basis states '0' or '1'. Quantum circuits operate in qubits which can assume values that are superposition of $|0>$ and $|1>$. Further, Quantum circuits are constrained networks of gates with no cloning and no feedback allowed [8].

A Quantum gate is a physical device implementing a unitary operator that represents the quantum state transformation.

III. Work Done in the Past

In general, so far, out of the four Evolutionary Algorithm types, only two i.e. Genetic Algorithms and Genetic Programming have been applied to the Quantum circuit synthesis. Genetic programming has been used to synthesize EPR (Einstein-Podolsky-Rosen) pairs of qubits, two-oracle AND/OR query problem [Spector99; Spector2006], database search problem [6], Teleportation [7] and Entanglement [10].

Quantum Teleportation being a truly Quantum phenomena, has been explored time and again by various research groups. Yabuki and Iba [9] have employed GA for evolving Teleportation circuit with lesser number of gates as compared to earlier circuits [2][5]. Another very important use of Genetic Algorithms has been in generation of Quantum equivalents for already existing classical circuits. It is known

that every quantum circuit is reversible [5], [8], so the researches on classical binary reversible synthesis and quantum synthesis share many ideas. The Reversible Logic (RL) circuits [1] are already technologically possible and have been implemented in CMOS technology [4]. An attempt at a general approach to encode both Quantum and Reversible Circuits was presented in Lukac et al. [11]. Quantum equivalent circuits for Binary Encoder, Decoder, Multiplexer, Half adder and full adder have been evolved using GA [18]. Younes and Miller [16] have presented an efficient technique by representing Quantum Boolean circuits using Reed-Muller expansion.

Shende et al. [13] proposed a top-down structure and effective computation by employing the Cosine-Sine Decomposition. With the help of an optimized quantum multiplexor, a quantum analog Shannon decomposition of Boolean functions is derived, by applying this decomposition recursively to quantum operators. This leads to a circuit synthesis algorithm in terms of quantum multiplexors.

IV. Genetic Algorithms for Designing Circuits

Perceiving the importance of GA in designing Quantum circuits, in this work, we explore the possibility of evolving an n qubit circuit when the circuit matrix has been provided using a set of single, two and three qubit gates.

The circuit model adopted in this paper is same as the one taken in [17].

A. Types of Gates

In the current coding scheme, 6 types of gates have been utilized viz. Hadamard, NOT, CNOT, SWAP, Toffoli and Fredkin where Hadamard and NOT are single qubit gates, CNOT and SWAP are two qubit gates and Toffoli and Fredkin are three qubit gates.

B. Initial population

Initial population with *m* chromosomes has been randomly generated. Each chromosome contains *p* number of gates but this size of chromosome, *p*, is fixed for the entire population. In the given problem, the population size has been taken as 150.

C. Fitness function

Once we have a population, we need to find how close it is to our desired output. Since target matrix is known, we define the fitness function as a matching percentage with the given output function by comparing each matrix element from our obtained output with each corresponding element from the expected solution. Thus, the fitness is calculated as:

$$F = count/total\ elements$$

Where *count* is the number of elements in obtained matrix matching exactly with their corresponding element in the desired matrix and total elements is the total number of elements in the matrix.

D. Selection process

After the fitness of each circuit is calculated, a number of selection procedures can be adopted for the generation of new population. In the work done by Debarati et al. [17], new population is obtained by crossover of all the chromosomes arranged in descending order and hence no particular selection process is adopted to pick parents which participate in crossover. On contrary, in our work we select the parents using Universal Stochastic selection in order to ensure that the fittest chromosome participates more often in the reproduction process for the next generation in line with the theory of survival of the fittest.

E. Crossover and Mutation

The k parents selected using the above mentioned procedure undergo crossover and mutation with the specified probabilities Pc=.7 and Pm=.05 respectively. Single point crossover has been performed. This is done by randomly choosing a crossover point in the first two parents, then next two and so on, and swapping all the gates that appear before that crossover point. At times a chromosome with high fitness value is selected as a parent more than once in the same generation. In this case, it is simply copy it to the next generation.

Once the crossover is performed mutation is done by randomly selecting a gate and replacing it with another gate in a chromosome. The probability of mutation is chosen to be small as compared to earlier work [17]. The effect of a low mutation rate on a population is that few variations are available to respond to sudden environmental change. This means the species is slower to adapt. On contrary, a higher mutation rate damages more individuals, though the population may be more adaptable to changing circumstances.

V. Evolving Simple Quantum Circuits

With the given set of gates, we have not only been able to evolve circuits for matrices provided by Debarati et al. [17] but also evolved these circuits with fewer gates and in much less time. In addition to this, we have been able to generalize this technique for evolving a circuit for any number of qubits using the given set of gates. Table I gives a comparison of the gate library used in their paper with ours.

TABLE I
Gate Library

	Gate Library used in [17]	Gate Library used in our work
Single qubit gate	Hadamard, Not	Hadamard, Not
Two qubit gate	CNOT	CNOT, Swap
Three qubit gate	Toffoli	TopToffoli MiddleToffoli, BottomToffoli, Fredkin

During the experiments it was observed that for each of the matrix given in the paper, there exists more than one possible solution circuit which was obtained during different runs of the GA. This was further verified by conducting an exhaustive search in the solution space. Table II illustrates the number of possible solutions (including the ones stated in [17]) for a given matrix along with a sample circuit with lesser number of gates.

VI. Quantum Equivalents of Boolean Circuits

As discussed earlier, unlike classical circuits, Quantum circuits are reversible in nature which makes it possible to retrieve the inputs values from the output values. Due to this property, Quantum circuits find an important application in the classical world; implementation of Boolean functions using Quantum circuits being one of them.

Younes and Miller in their work [14] have introduced the technique for representing Quantum Boolean circuits using Reed-Muller expansion. They have mainly focused on generalized CkNOT based circuit synthesis. The work by Amlan et al. [15] focuses on defining the synthesis technique for Boolean circuits utilizing nearest neighbor template for CNOT and C2NOT gates. More recently, Lukac et al. have developed a parallel Genetic Algorithm that synthesizes Boolean reversible circuits realized with a variety of Quantum gates on qudits with various radices.

In our work, we have tried to evolve Quantum equivalent circuits for 2 and 3 bit Boolean functions using our simple Genetic Algorithm described in previous sections.

A. Scratch Bits and Ancillas

Just as in classical systems, circuits can be made more efficient by using some extra bits called the 'scratch bits' which can be set to zero for use and then be discarded, in order to calculate the Boolean function using a Quantum circuit, we make use of this scratch bit called 'ancilla qubit' by introducing it at the input with value zero and then later on storing the function value into it.

So for instance, the circuit for a 2-qubit function $F(x_1,x_2)$, receives x_1 and x_2 as the first two (actual) inputs and a third F_{in} with value zero. At the output, this becomes F_{out} and contains the value of the function.

B. Oracles

Similar to classical computation where an oracle is a black-box with *n*-bit number x as input and function F_x as output, the idea of oracle plays a vital role in Quantum computation. In the case of an oracle is a black box which takes n-qubits and performs a unitary transformation $\hat{U}$ (which could in turn be a product of transformations) on them.

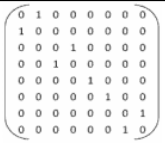
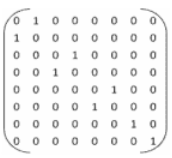
C. Evolving Boolean Functions

In order to find a Quantum circuit for a given Boolean function, the following 2 steps need to be followed.

- Using Karnaugh maps, we calculate the binary matrix corresponding to the target Boolean function. This serves as the unitary transformation which acts on the inputs to give the desired output.
- Once this target matrix is obtained, we implement GA (as discussed earlier sections) to find the Quantum gates that compose the circuit yielding the unitary transformation representing the oracle.

Some of the Boolean functions that have been successfully evolved by have been listed in the Table III. The gate library constitutes NOT, CNOT, Toffoli and Fredkin gates. The reason for selecting this set of gates is that they represent a very basic standard of well known and used universal reversible gates. Further, here again it was observed that different runs of the algorithm gave different solutions for the same problem. Hence we could obtain more than one solution for several circuits, with the same as well as different number of gates.

TABLE III
Quantum Equivalent Circuits For Boolean Functions

Boolean Function	Target matrix	No. of gates	Circuit obtained
$F(x_1, x_2) = \overline{x}_1 + x_2$		3	CNOT (1,3) NOT 3 TOFFOLI (1,2,3)
		4	CNOT (1,2) FREDKIN (3,1,2) NOT 3 TOFFOLI (1,3,2)
$F(x_1, x_2) = x_1 x_2$		4	CNOT (2,1) CNOT (2,3) TOFFOLI (1,2,3) CNOT (2,1)
		6	FREDKIN (1,2,3) NOT 3 TOFFOLI (1,3,2) CNOT (1,2) NOT 3 FREDKIN (1,2,3)
$F(x_1, x_2) = x_1\overline{x}_2 + \overline{x}_1$		4	CNOT (1,3) TOFFOLI (1,2,3) CNOT (1,3) NOT 3
		5	CNOT (2,1) NOT 3 TOFFOLI (1,2,3) CNOT (2,1) CNOT (2,3)

VII. Conclusion

Designing a quantum circuit to solve a given problem is not simple because of the little knowledge about a search space we are dealing with. It is not only difficult to predict the effect of local change on the circuit property but there is also no yardstick for judging its efficiency. Thus, the circuits created manually are prone to human errors and take a lot of time in development.

Through our research work we have proved the effectiveness of Genetic Algorithms in evolving different kinds of circuits if the corresponding circuit matrices have been given. Further, through universal stochastic sampling procedure we have been able to obtain desired results much

faster as opposed to conventional unstructured processes stated in earlier researches, thereby making the systems more efficient. Not only are the circuits effective and efficient, they are much smaller in size as compared to earlier circuits. In addition to gain in speed and size, we have also obtained more than one solution for the same problem enabling us to optimize our solution. These aspects can be probed further to evaluate commercial and industrial cost savings. In another interesting application of GA, it has been observed that this algorithm can be instrumental in evolving Boolean functions for Quantum computers. Initial experiments successfully helped in creating classical Boolean circuits using Quantum gates. In future, this method can be implemented to obtain Boolean operations such as addition, multiplication and functions like multiplexers, encoders etc. We propose to present the findings of this research work in the subsequent papers.

TABLE II
COMPARISON OF CIRCUITS OBTAINED IN THIS WORK AND IN [17]

No. of qubits	Target matrix	Circuit obtained by Debarati	No. of gates	No. of gener-ations	Time taken (in secs.)	Circuit obtained by us using Debarati library	No. of gates	No. of gener-ations	Time taken (in secs.)	Total no. of possible solutions	Circuit obtained using our extended library
2	0.7071 0.7071 0 0 0 0 0.7071 0.7071 -0.7071 0.7071 0 0 0 0 0.7071 -0.7071	HADAMARD 1 NOT 1 CNOT (1,2) CNOT (2,1)	4	6	48	HADAMARD 1 CNOT (1,2) CNOT (2,1) NOT 2	4	9	.875	2	HADAMARD 1 NOT 1 SWAP(1,2) CNOT (1,2)
		HADAMARD 1 NOT 2 CNOT (1,2) CNOT(2,1) NOT 2 NOT 1 NOT2	7	13	65	HADAMARD 1 CNOT (1,2) NOT 2 NOT 1 CNOT (2,1)	5	3	.66	7	SWAP HADAMARD 2 CNOT (1,2) NOT 2
3	0.5 0 0.5 0 0.5 0 0.5 0 0 0.5 0 0.5 0 0.5 0 0.5 0.5 0 -0.5 0 0.5 0 -0.5 0 0 -0.5 0 0.5 0 -0.5 0 0.5 0 0 0.5 0.5 0 0 -0.5 -0.5 0.5 0.5 0 0 -0.5 -0.5 0 0 0 0 0.5 -0.5 0 0 -0.5 0.5 -0.5 0.5 0 0 0.5 -0.5 0 0	HADAMARD 2 TOFFOLI CNOT (1,2) HADAMARD 1 CNOT (3,2)	5	11	110	HADAMARD 2 TOFFOLI CNOT (3,2) CNOT (1,2) HADAMARD 1	5	45	14.1	3	HADAMARD 2 CNOT (3,2) FREDKIN CNOT (1,2) HADAMARD 1
		HADAMARD 2 TOFFOLI CNOT (1,2) CNOT (3,2) HADAMARD 1	5	30	248	HADAMARD 2 TOFFOLI NOT 1 CNOT (1,2) NOT 1 CNOT (3,2) NOT 2 HADAMARD 1	8	443	230.7	2	
3	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0	CNOT (2,1) CNOT (2,1) TOFFOLI CNOT (2,1) CNOT (3,2) NOT 1	6	17	136	TOFFOLI NOT 1 CNOT (2,1) CNOT (3,2)	4	8	1.3	8	CNOT (2,1) CNOT (3,2) NOT 1 FREDKIN
		TOFFOLI CNOT (2,3) CNOT (2,1) CNOT (2,3) CNOT (3,2) NOT 1	6	2	20	CNOT (2,1) CNOT (2,3) TOFFOLI CNOT (3,2) NOT 1	5	11	2.4	6	CNOT (3,1) NOT 1 CNOT (3,2) CNOT (2,1) FREDKIN
4	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	NOT 2 NOT 4 CNOT (3,2) NOT 3 NOT 1 CNOT (3,2) CNOT (3,4) NOT 1 CNOT (4,3)	9	20	390	CNOT (3,4) NOT 3 CNOT (4,3)	3	9	3.38	2	
		CNOT (2,1) CNOT (4,3) CNOT (2,1) CNOT (4,3) NOT 2 NOT 2 CNOT (3,2) NOT 1 CNOT (3,2)	9	36	555	CNOT (3,4) NOT 4 CNOT (4,3) NOT 4	4	2	.99	5	NOT 4 CNOT (3,4) SWAP (3,4) NOT 4

REFERENCES

[1] E. Fredkin and T. Toffoli, "Conservative Logic", International Journal of Theoretical Physics, No. 21, pp. 219-253, 1982
[2] G. Brassard, S.L. Braunstein, and R. Cleve, "Teleportation as a quantum computation", Physica D: Nonlinear Phenomena, 120(1-2):43{47, 1998.
[3] C. P. Williams and S. H. Clearwater, "Explorations in quantum computing". Springer- Verlag, TELNOS, 1997.
[4] C. Veiri, A. Josephine and M. Frank, "A fully Reversible Asymptotically Zero Energy Microprocessor", MIT AI Laboratory, 1998
[5] C. P. Williams and S. H. Clearwater, "Exploration in Quantum Computing", Springer-Verlag, 1998.
[6] L. Spector, H. Barnum, H. J. Bernstein, and N. Swamy, "Quantum computing applications of genetic programming". Advances in Genetic Programming, 1999.
[7] C. P. Williams and G. G. Alexander, "Automated design of quantum circuits", QCQC'98, LNCS, 1999.
[8] M. Nielsen and I. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, (2000)
[9] T. Yabuki and H. Iba, "Genetic algorithms for quantum circuit design - Evolving a simpler teleportation circuit". In Late Breaking Papers at GECCO 2000.
[10] B.I.P. Rubinstein, "Evolving quantum circuits using genetic programming", Proceedings of the 2001 Congress on Evolutionary Computation, 2001
[11] M. Lukac and M. Perkowski, "Evolving quantum circuits using genetic algorithm", Proceedings of th1e 2002 NASA/DOD Conference on Evolvable Hardware, 2002.
[12] M. Lukac, M. Perkowski, H. Goi, M. Pivtoraiko, C. H. Yu, K. Chung, H. Jee, B. G. Kim and Y. D. Kim, "Evolutionary Approach To Quantum And Reversible Circuits Synthesis", 2003
[13] V.V. Shende, S. S. Bullock, I. L. Markov, "Synthesis of Quantum Logic Circuits", Quantum Physics, quant-ph/0406176, 2004
[14] A. Younes and J. Miller, "Representation of Boolean Quantum Circuits as Reed-Muller Expansions", International Journal of Electronics, Vol. 91, No. 7, 2004
[15] A. Chakrabarti and S. S. Kolay, "Rules for Synthesizing Quantum Boolean Circuits Using Minimized Nearest Neighbor Templates", 15th International Conference on Advanced Computing and Communication, 2007
[16] A. Younes and J. Miller, "Automated Method for Building CNOT based Quantum Circuits for Boolean Functions"
[17] D. Mukherjee, A. Chakrabarti, D. Bhattacherjee, "Synthesis of Quantum Circuits Using Genetic Algorithm", International Journal of Recent Trends in Engineering, Vol 2, No. 1, November 2009
[18] D. Mukhopadhyay and A. Si, "Quantum Multiplexer Desigining and Optimization applying Genetic Algorithm", International Journal of Computer Science, Vol. 7, Issue 5, 2010