13. Zhang JJ, Zhang T. Parameter-induced stochastic resonance based on spectral entropy and its application to weak signal detection. Review of Scientific Instruments. 2015;86(2):6.

14. He MJ, Xu W, Sun ZK, Du L. Characterization of stochastic resonance in a bistable system with Poisson white noise using statistical complexity measures. Communications in Nonlinear Science and Numerical Simulation. 2015;28(1-3):39-49.

15. Liu SJ, Yang T, Zhang XZ. Effects of stochastic resonance for linear-quadratic detector. Chaos Solitons & Fractals. 2015;77:319-31.

Digital Circuit Optimization Design Algorithm Based on Cultural Evolution

Xuesong Yan[1], Qinghua Wu[2,3], Hanmin Liu[4]

[1]School of Computer Science, China University of Geosciences,
Wuhan, Hubei, 430074, China
[2]Hubei Provincial Key Laboratory of Intelligent Robot, Wuhan Institute of Technology,
Wuhan, Hubei, 430073, China
[3]School of Computer Science and Engineering, Wuhan Institute of Technology,
Wuhan, Hubei, 430073, China
[4]Wuhan Institute of Ship Building Technology, Wuhan, Hubei, 430050
**Corresponding author*

Corresponding author is Qinghua Wu

Abstract
Cultural Algorithms (CA) are a class of computational models derived from observing the cultural evolution process in nature. The Cultural Algorithm is a dual inheritance system that characterizes evolution in human culture at both the macro-evolutionary level, which takes place within the belief space, and at the micro-evolutionary level, which occurs at the population space. Cultural algorithm is used to solve complex calculations of the new global optimization search algorithms, cultural algorithms in the optimization of the complex functions of its superior performance. In this paper, we use the this algorithm for combinational digital circuit optimization design, from the experiment results shown our algorithm are effective for this problem.

Keywords: COMBINATIONAL DIGITAL CIRCUIT, CULTURAL ALGORITHM, CIRCUIT OPTIMIZATION DESIGN, CIRCUIT EVALUATION

1. Introduction

Evolutionary Electronics applies the concepts of genetic algorithms to the evolution of electronic circuits. The main idea behind this research field is that each possible electronic circuit can be represented as an individual or a chromosome of an evolutionary process, which performs standard genetic operations over the circuits. Due to the broad scope of the area, researchers have been focusing on different problems, such as placement, Field Programmable Gate Array (FPGA) mapping, optimization of combinational and sequential digital circuits, synthesis of digital circuits, synthesis of passive and active analog circuits, synthesis of operational amplifiers, and transistor size optimization. Of great relevance are the works focusing on “intrinsic” hardware evolution in which fitness evaluation is performed in silicon, allowing a higher degree of exploration of the physical properties of the medium. This particular area is frequently called Evolvable Hardware [1-3].

In the sequence of this work, Coello, Christiansen and Aguirre presented a computer program that automatically generates high-quality circuit designs [4]. Miller, Thompson and Fogarty applied evolutionary algorithms for the design of arithmetic circuits [5]. Kalganova, Miller and Lipnitskaya proposed another technique for designing multiple-valued circuits [6]. In order to solve complex systems, Torresen proposed the method of increased complexity evolution. The idea is to evolve a system gradually as a kind of divide-and-conquer method [7]. Based on the Miller’s method, Yan applied Gene Expression Programming (GEP) [8, 9], Particle Swarm Optimization Algorithms (PSO) [10-14], Cultural Algorithms (CA) [15-20], Orthogonal Evolutionary Algorithm [21] and evolutionary algorithm [22-24] for the design of electronic circuits.

A major bottleneck in the evolutionary design of electronic circuits is the problem of scale. This refers to the very fast growth of the number of gates, used in the target circuit, as the number of inputs of the evolved logic function increases. This results in a huge search space that is difficult to explore even with evolutionary techniques. Another related obstacle is the time required to calculate the fitness value of a circuit. A possible method to solve this problem is to use building blocks either than simple gates. Nevertheless, this technique leads to another difficulty, which is how to define building blocks that are suitable for evolution.

2. Cultural algorithm

Evolutionary computation (EC) [25, 26] methods have been successful in solving many diverse problem in search and optimization due to the unbiased nature of their operations which can still perform well in situation with little or no domain knowledge. However, there can be considerable improvement in their performance when problem specific knowledge is used to bias the problem solving process in order to identify patterns in their performance environment. These patterns are used to promote more instances of desirable candidates or to reduce the number of less desirable candidates in the population. In either case, this can afford the system an opportunity to reach the desired solution more quickly.

Adaptive evolutionary computation takes place when an EC system is able to incorporate such information into its representation and operators in order to facilitate the pruning and promoting activities mentioned above. Some research works have shown that self-adaptation can take place on several levels within a system such as the population level, the individual level, and the component level. At the population level, aspects of the system parameters that control all elements of the population can be modified. At the individual level, aspects of the system that control the action of specific individual can be modified. If the individual is specified as s collection of components then component level adaptation is possible. This involves the adaptation of parameters that control the operation of one or more components that make up an individual.

Several challenges arise in optimization. First is the nature of the problem to be optimized which may have several local optima the optimizer can get stuck in, the problem may be discontinuous, candidate solutions may yield different fitness values when evaluated at different times, and there may be constraints as to what candidate solutions are feasible as actual solutions to the real-world problem. Furthermore, the large number of candidate solutions to an optimization problem makes it intractable to consider all candidate solutions in turn, which is the only way to be completely sure that the global optimum has been found. This difficulty grows much worse with increasing dimensionality, which is frequently called the curse of dimensionality, a name that is attributed to Bellman, see for example [27]. This phenomenon can be understood by first considering an n-dimensional binary search-space. Here, adding another dimension to the problem means a doubling of the number of candidate solutions. So the number of candidate solutions grows exponentially with increasing dimensionality. The same principle holds for continuous or real-valued search-spaces, only it is now the volume of the search-space that grows exponentially with increasing dimensionality. In either case it is therefore of great interest to find optimization methods which not only perform well in few dimensions, but do not require an exponential number of fitness evaluations as the dimensionality grows. Preferably such optimization methods have a linear relationship between the dimensionality of the problem and the number of candidate solutions they must evaluate in order to achieve satisfactory results, that is, optimization methods should ideally have linear time-complexity O(n) in the dimensionality n of the problem to be optimized.

Another challenge in optimization arises from how much or how little is known about the problem at hand. For example, if the optimization problem is given by a simple formula then it may be possible to derive the inverse of that formula and thus find its optimum. Other families of problems have had specialized methods developed to optimize them efficiently. But when nothing is known about the optimization problem at hand, then the No Free Lunch (NFL) set of theorems by Wolpert and Macready states that any one optimization method will be as likely as any other to find a satisfactory solution [28]. This is especially important in deciding what performance goals one should have when designing new optimization methods, and whether one should attempt to devise the ultimate optimization method which will adapt to all problems and perform well. According to the NFL theorems such an optimization method does not exist and the focus of this thesis will therefore be on the opposite: Simple optimization methods that perform well for a range of problems of interest.

Cultural Algorithms (CA) are a class of computational models derived from observing the cultural evolution process in nature [29, 30]. The Cultural Algorithm is a dual inheritance system that characterizes evolution in human culture at both the macro-evolutionary level, which takes place within the belief space, and at the micro-evolutionary level, which occurs at the population space. CA consists of a social population and a belief space. Experience of individuals selected from the population space by the acceptance function is used to generate problem solving knowledge that resides in the belief space. The belief space stores and manipulates the knowledge acquired from the experience of individuals in the population space. This knowledge can control the evolution of the population component by means of the influence function. As a result, CA can provide an explicit mechanism for global knowledge and a useful framework within which to model self-adaptation in an EC system. The population level component of the cultural algorithm will be Evolutionary Programming (EP). The global knowledge that has been learned by the population will be expressed in terms of both normative and situational knowledge as discussed earlier.

A flowchart of the Cultural Algorithms as shown in Figure 1.

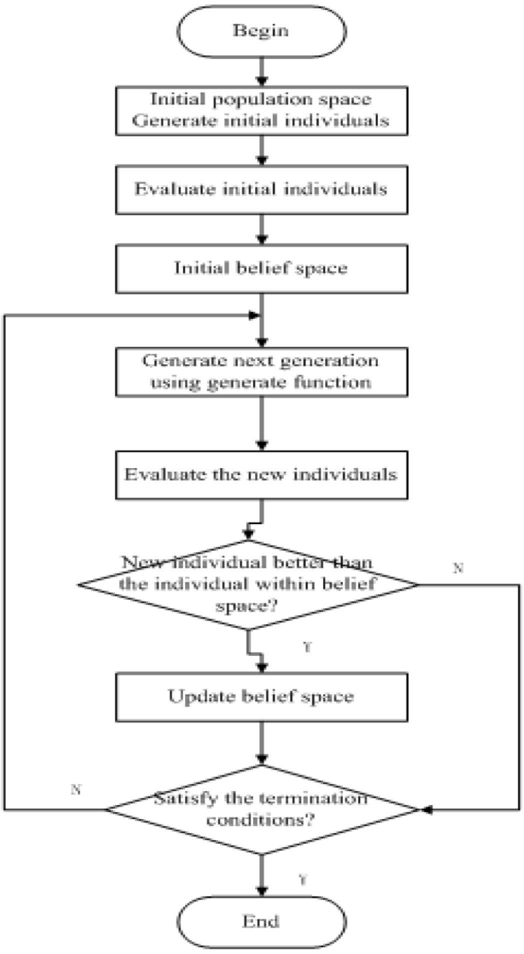


Figure 1. Flowchart of cultural algorithm

In this algorithm, first the belief space and the population space are initialized. Then, the algorithm will repeat processing for each generation until a termination condition is achieved. Individuals are evaluated using the performance function. The two levels of Cultural Algorithm communicate through the acceptance function and the influence function. The acceptance function determines which individuals from the current population are selected to impact the belief space. The selected individuals’ experiences are generalized and applied to adjust the current beliefs in the belief space via the update function. The new beliefs can then be used to guide and influence the evolutionary process for the next generation. Cultural algorithms as described above consist of three components. First, there is a population component that contains the social population to be evolved and the mechanisms for its evaluation, reproduction, and modification. Second there is a belief space that represents the bias that has been acquired by the population during its problem-solving process. The third component is the communications protocol that is used to determine the interaction between the population and their beliefs.

Cultural algorithm is in-depth analysis of the superiority of the original evolution theory on the basis of drawing on the social (cultural) evolution theory in the social sciences and has achieved broad consensus on the research results, and proposed a new algorithm. Cultural algorithm is used to solve complex calculations of the new global optimization search algorithms, cultural algorithms in the optimization of the complex functions of its superior performance.

3. Combinational digital circuit representation

3.1. Circuit coding

In this paper, the chromosome representation we use Miller's [5]. This representation is based on the Field Programming Gate Array (FPGA) of Xilinx Virtex-II (Figure 2 is the structure of FPGA). The starting point in this technique is to consider, for each potential design, a geometry (of a fixed size array) of uncommitted logic cells that exist between a set of desired inputs and outputs. The uncommitted logic cells are typical of the resource provided on the Xilinx FPGA part under consideration. An uncommitted logic cell refers to a two-input, single-output logic module with no fixed functionality. The functionality may then be chosen, at the implementation time, to be any two input variable logic function. In this technique, a chromosome is defined as a set of interconnections and gate level functionality for these cells from outputs back toward the inputs based upon a numbered rectangular grid of the cells themselves, as in Figure 3. The inputs that are made available are logic '0', logic '1', all primary inputs and primary inputs inverted. To illustrate this consider a 3 x 3 array of logic cells between two required primary inputs and two required outputs.

The inputs 0 and 1 are standard within the chromosome, and represent the fixed values, logic '0' and logic '1' respectively. The inputs (two in this case) are numbered 2 and 3, with 2 being the most significant. The lines 4 and 5 represent the inverted inputs 2 and 3 respectively. The logic cells which form the array are numbered column-wise from 6 to 14. The outputs are numbered 13 and 11, meaning that the most significant output is connected to the output of cell 13 and the least significant output is connected to the output of cell 11. These integer values, whilst denoting the physical location of each input, cell or output within the structure, now also represent connections or routes between the various points. In other words, this numbering system may be used to define a netlist for any combinational circuit. Thus, a chromosome is merely a list of these integer values, where the position on the list tells us the cell or output which is being referred to, while the value tells us the connection (of cell or input) to which that point is connected, and the cells functionality.

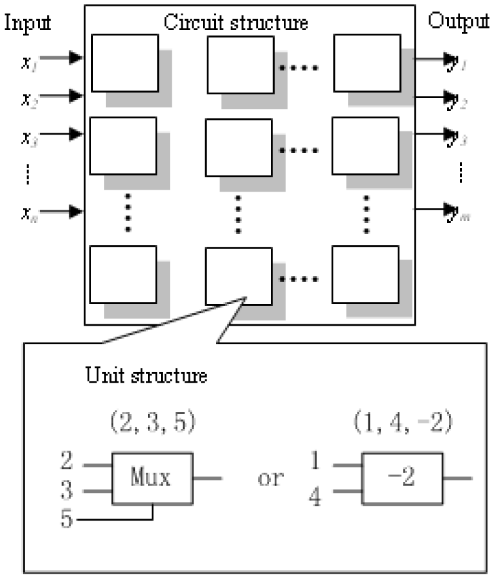


Figure 2. Structure of FPGA

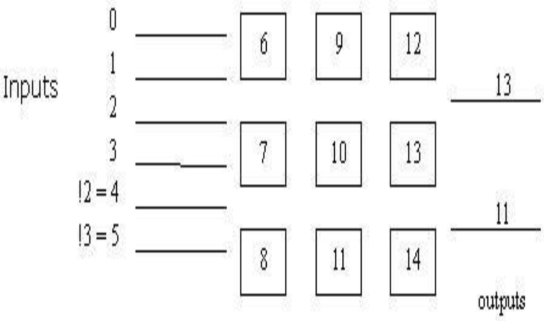
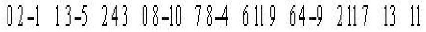


Figure 3. A 3 x 3 geometry of uncommitted logic cells with inputs, outputs and netlist numbering

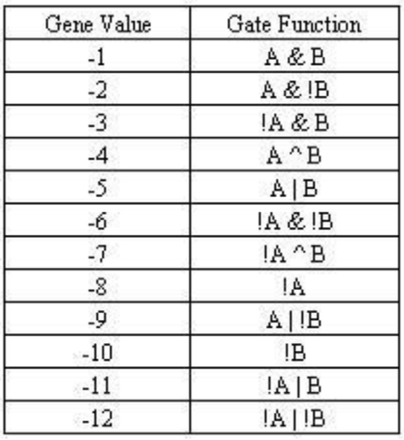
Each of the logic cells is capable of assuming the functionality of any two-input logic gate, or, alternatively a 2-1 multiplexer (MUX) with single control input. In the geometry shown in Figure 3, a sample chromosome is shown below:



0 2 -1 1 3 -5 2 4 3 0 8 -10 7 8 -4 6 11 9 6 4 -9 2 11 7 13 11

Figure 4. A typical netlist chromosome for the 3 x 3 geometry of Figure 3

Notice, in this arrangement that the chromosome is split up into groups of three integers. The first two values relate to the connections of the two inputs to the gate or MUX. The third value may either be positive - in which case it is taken to represent the control input of a MUX - or negative - in which case it is taken to represent a two-input gate, where the modulus of the number indicates the function according to Figure 5 below. The first input to the cell is called A and the second input called B for convenience. For the logic operations, the C language symbols are used: (i) & for AND, (ii) | for OR, (iii) ^ for exclusive-OR, and (iv) ! for NOT. There are only 12 entries on this table out of a possible 16 as 4 of the combinations: (i) all zeroes, (ii) all ones, (iii) input A passed straight through, and (iv) input B passed straight through are considered trivial - because these are already included among the possible input combinations, and they do not affect subsequent network connections in cascade.



Gene Value	Gate Function
-1	A & B
-2	A & !B
-3	!A & B
-4	A ^ B
-5	A | B
-6	!A & !B
-7	!A ^ B
-8	!A
-9	A | !B
-10	!B
-11	!A | B
-12	!A | !B

Figure 5. Cell gate functionality according to negative gene value in chromosome

This means that the first cell with output number 6 and characterized by the triple {0, 2, -1} has it's a input connected to '0', its B input connected to input 2, and since the third value is -1, the cell is an AND gate (thus in this case always produces a logical output of 0). Picking out the cell who's output is labeled 9, which is characterized by the triple {2, 6, 7}, it can be seen that its A input is connected to input 2 and its B input is connected to the output of cell 6, while since the third number is positive the cell is a MUX with control input connected to the output of cell 7.

3.2. Circuit evaluation

In practical applications, the evaluation of the quality of a circuit, in addition to function correctly, there are two important parameters: the delay time and power consumption. The evolution of digital circuit design process, even if the correct solution, but the circuit is not necessarily the optimal circuit functions correctly premise, the delay is the most important indicators of the evaluation circuit is good or bad, secondly, the power consumption is also to reflect the performance of the circuit important indicator of the evolution of the design of digital circuits has three goals descending order of importance: the correct function, minimize delay time and minimum power consumption. Meet the three objectives of the circuit is the best circuit.

For a specific circuit, as shown in Figure 6, there may be a plurality of output terminals, there may be many different paths from the input to the output, the delay of each path are not normally equal, we calculated each path delay, as to which the maximum delay of the entire circuit, such as the figure 8 above, a full adder, from the A input or the B input, through XOR gate, AND gate to reach the OR gate, then to C0, this road of maximum delay path and calculate the delay time as follows:

T =Tinstrinsic(XOR) + Tinstrinsic(AND) + + Tinstrinsic(OR)

First XOR: t1= 0.1373(ns)

Second AND: t2= 0.0801(ns)

Third OR: t3= 0.0617(ns)

Total delay tiem: t= t1 + t2 + t3 = 0.2791(ns)

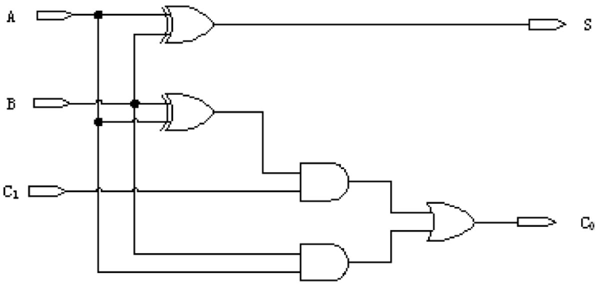


Figure 6. One-bit adder circuit

Calculate the power consumption is simple, to accumulate all logic gates used in the circuit power consumption. So the power consumption of the Figure 8 as:

W = W (XOR) * 2 + W (AND) * 2 + W (OR) = = 0.0372 * 2 + 0.0103 * 2 + 0.0191 = 0.1141

According to the design requirements of the circuit gate count, power consumption, delay, and the fitness function of the circuit as follows:

$$G(x) = \begin{cases} F(x) & F < 2^n \quad (circuit \quad wrong) \\ F(x) + w * V(x)) & F = 2^n \quad (circuit \quad correct) \end{cases}$$

In here, *F(x)* is the circuit x full output cell information and the truth table of the degree of matching, *V(x)* for x is the effective number of gates of the circuit, the circuit is completely correct conditions, the smaller the effective number of gates, the better the circuit performance. *w* is the weight value factor, taking a very small positive number.

4. Circuit design case

Evolving the one-bit adder was easier to do on a larger geometry but resulted in a less efficient circuit. That is many genetic algorithm was able to discover 100% functional solutions was intimately related to the size of the geometry, but our algorithm use small geometry to find the fully functional solutions. Figure 7 is the circuit designed by CA (with three gates) and Figure 8 is the circuit designed by Miller [31]. From the figure we know it is a gratifying result to obtain as it is clear that this design is an optimum solution.

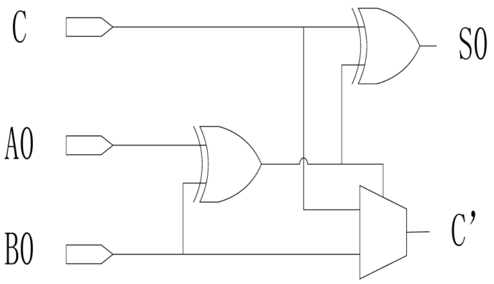


Figure 7. One-bit full adder circuit designed by CA

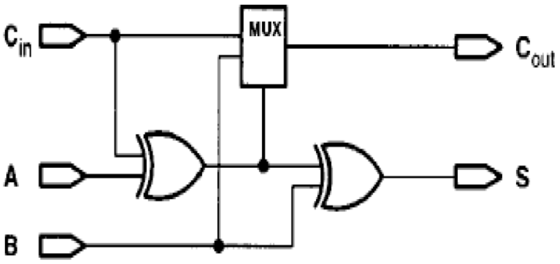


Figure 8. One-bit full adder circuit designed by Miller

A two-bit full adder circuit, which with a truth table with 5 inputs and 3 outputs. In this case, Our algorithm use small geometry to find the fully functional solutions, the matrix has a size of 3×3. The resulting circuits as shown in Figure 9 (with six gates) and Figure 10 is the circuit designed by Miller [31]. From the figures we know it is a gratifying result to obtain as it is clear that this design is an optimum solution.

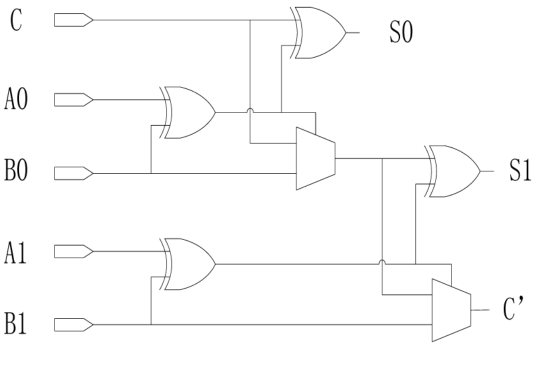


Figure 9. Two-bit full adder circuit designed by CA

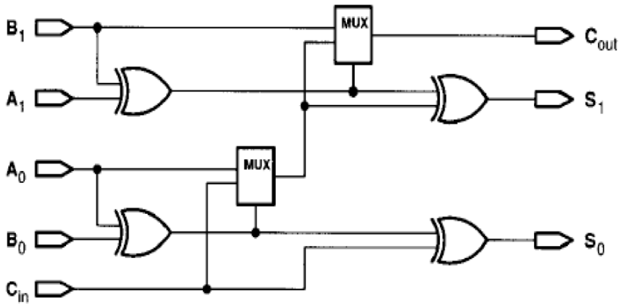


Figure 10. Two-bit full adder circuit designed by Miller

We also use our algorithm to evolve the three-bit full adder and four-bit full adder. A three-bit full adder, with a truth table with 7 inputs and 4 outputs. In this case, our algorithm use small geometry to find the fully functional solutions, the matrix has a size of 4×4. The resulting circuits as shown in Figure 11 (with 9 gates). From the figures we know it is a gratifying result to obtain as it is clear that this design is an optimum solution.

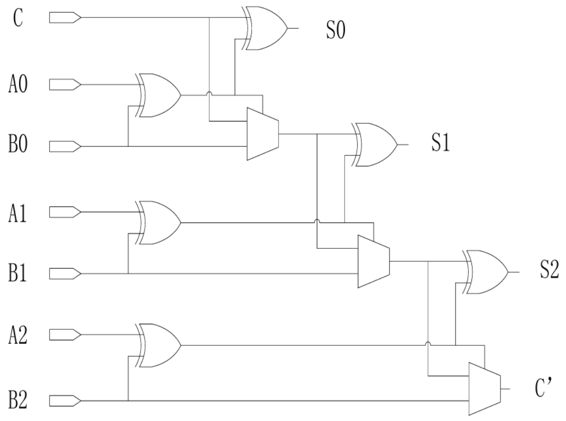


Figure 11. Three-bit full adder circuit designed by CA

A four-bit full adder, with a truth table with 9 inputs and 5 outputs. In this case, our algorithm use small geometry to find the fully functional solutions, the matrix has a size of 8×8. The resulting circuits as shown in Figure 12 (with 12 gates). From the figures we know it is a gratifying result to obtain as it is clear that this design is an optimum solution.

We run our circuit design algorithm 10 times for the four circuit design problems and verify the algorithm's effective. Figure 13 are the experiment results of two-bit full adder among 10 runs. In Figure 14 are the experiment results for the four circuit design cases.

5. Conclusion

In human societies, culture can be a vehicle for the storage of information in a form that is independent of the individual or individuals that generated and are potentially accessible to all members of the society.

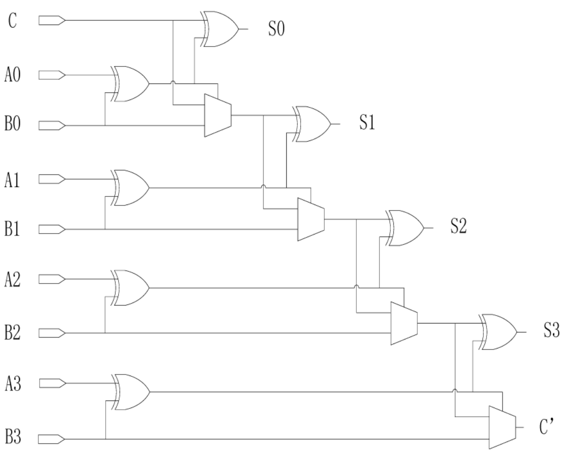
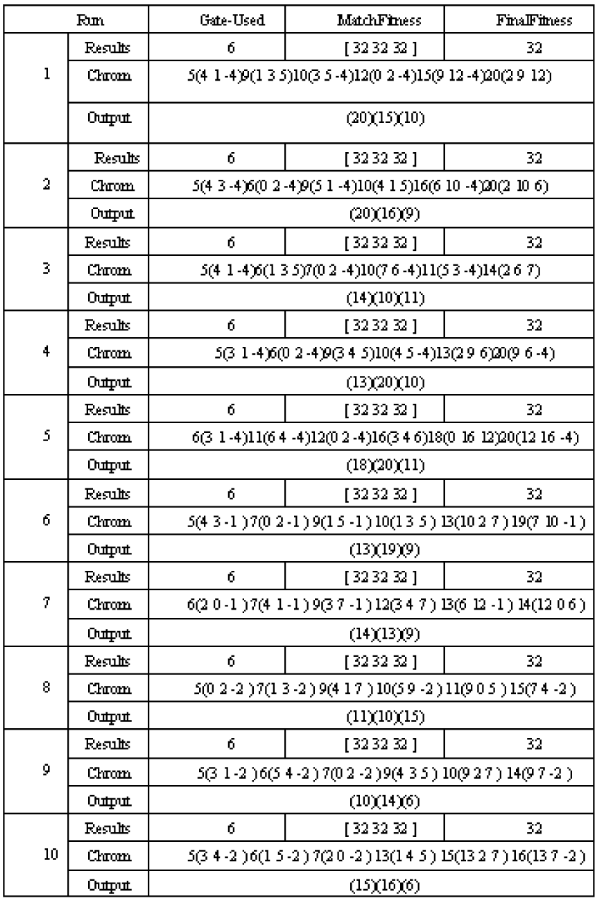
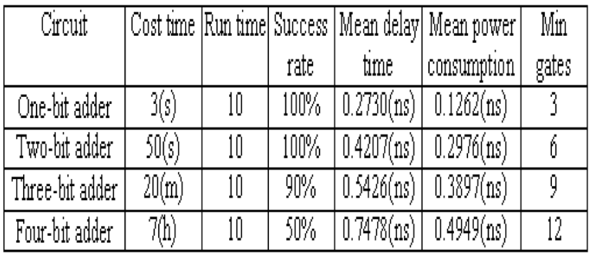


Figure 12. Four-bit full adder circuit designed by CA



Run		Gate-Used	MatchFitness	FinalFitness
1	Results	6	[32 32 32]	32
	Chrom	5(4 1 -4)9(1 3 5)10(3 5 -4)12(0 2 -4)15(9 12 -4)20(2 9 12)		
	Output	(20)(15)(10)		
2	Results	6	[32 32 32]	32
	Chrom	5(4 3 -4)6(0 2 -4)9(5 1 -4)10(4 1 5)16(6 10 -4)20(2 10 6)		
	Output	(20)(16)(9)		
3	Results	6	[32 32 32]	32
	Chrom	5(4 1 -4)6(1 3 5)7(0 2 -4)10(7 6 -4)11(5 3 -4)14(2 6 7)		
	Output	(14)(10)(11)		
4	Results	6	[32 32 32]	32
	Chrom	5(3 1 -4)6(0 2 -4)9(3 4 5)10(4 5 -4)13(2 9 6)20(9 6 -4)		
	Output	(13)(20)(10)		
5	Results	6	[32 32 32]	32
	Chrom	6(3 1 -4)11(6 4 -4)12(0 2 -4)16(3 4 6)18(0 16 12)20(12 16 -4)		
	Output	(18)(20)(11)		
6	Results	6	[32 32 32]	32
	Chrom	5(4 3 -1)7(0 2 -1) 9(1 5 -1) 10(1 3 5) 13(10 2 7) 19(7 10 -1)		
	Output	(13)(19)(9)		
7	Results	6	[32 32 32]	32
	Chrom	6(2 0 -1)7(4 1 -1) 9(3 7 -1) 12(3 4 7) 13(6 12 -1) 14(12 0 6)		
	Output	(14)(13)(9)		
8	Results	6	[32 32 32]	32
	Chrom	5(0 2 -2)7(1 3 -2) 9(4 1 7) 10(5 9 -2) 11(9 0 5) 15(7 4 -2)		
	Output	(11)(10)(15)		
9	Results	6	[32 32 32]	32
	Chrom	5(3 1 -2)6(5 4 -2) 7(0 2 -2)9(4 3 5) 10(9 2 7) 14(9 7 -2)		
	Output	(10)(14)(6)		
10	Results	6	[32 32 32]	32
	Chrom	5(3 4 -2)6(1 5 -2) 7(2 0 -2) 13(1 4 5) 15(13 2 7) 16(13 7 -2)		
	Output	(15)(16)(6)		

Figure 13. Two-bit full adder results among 10 runs



Circuit	Cost time	Run time	Success rate	Mean delay time	Mean power consumption	Min gates
One-bit adder	3(s)	10	100%	0.2730(ns)	0.1262(ns)	3
Two-bit adder	50(s)	10	100%	0.4207(ns)	0.2976(ns)	6
Three-bit adder	20(m)	10	90%	0.5426(ns)	0.3897(ns)	9
Four-bit adder	7(h)	10	50%	0.7478(ns)	0.4949(ns)	12

Figure 14. Experiment results of full-adder circuit

As such culture is useful in guiding the problem solving activities and social interaction of individuals in the population. This allows self-adaptive information as well as other knowledge to be stored and manipulated separately from the individuals in the social population. This provides a systematic way of utilizing self-adaptive knowledge to direct the evolution of a social population. Cultural Algorithms in order to model the evolution of cultural systems based upon principles of human social evolution taken from the social science literature. This paper we use the this algorithm for combinational digital circuit optimization design. Compared with the traditional circuit design algorithm, this algorithm enlarges the searching space and the complexity is not high. For the circuit design cases studies this means has proved to be efficient, experiments show that we have better results than the traditional algorithm.

Acknowledgements

This paper is supported by Natural Science Foundation of China. (No.61272470, 61203307, 61440060 and 41404076), the Provincial Natural Science Foundation of Hubei (2015CFA065, 2013CFB320).

References

1. Zebulum, R. S., Pacheco, M. A. and Belasco, M. M. Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms. CRC Press, 2001.
2. Thompson, A. and Layzell, P. Analysis of unconventional evolved electronics. Communications of the ACM, 1999, Vol. 42, pp.71-79.
3. Louis, S.J. and Rawlins, G. J. Designer Genetic Algorithms: Genetic Algorithms in Structure Design. Proceedings of the Fourth International Conference on Genetic Algorithms, 1991, pp. 53-60.
4. Cello, C. A., Christiansen, A. D. and Aguirre, A. H. Using Genetic Algorithms to Design Combinational Logic Circuits. Intelligent Engineering through Artificial Neural Networks, 1996, Vol. 6, pp. 391-396.
5. Miller, J. F., Thompson, P. and Fogarty, T. Algorithms and Evolution Strategies in Engineering and Computer Science: Recent Advancements and Industrial Applications. 1997.
6. Kalgan ova, T., Miller, J. F. and Lipnitskaya, N. Multiple-Valued Combinational Circuits Synthesised using Evolvable Hardware. Proceedings of the 7th Workshop on Post-Binary Ultra Large Scale Integration Systems, 1998.

7. Torresen, J. A Divide-and-Conquer Approach to Evolvable Hardware. Proceedings of the Second International Conference on Evolvable Hardware, 1998, Vol. 1478, pp. 57-65.
8. X.S.Yan, Wei Wei. Design Electronic Circuits by Means of Gene Expression Programming. Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems, 2006, pp. 194-199.
9. X.S.Yan, Wei Wei. Designing Electronic Circuits by Means of Gene Expression Programming Ⅱ. Proceedings of the 7th International Conference on Evolvable Systems: From Biology to Hardware, 2007, pp. 319-330.
10. X.S. Yan, Qinghua Wu. Circuit Design Based on Particle Swarm Optimization Algorithms. Key Engineering Materials, 2011, Vols. 474-476, pp. 1093-1098.
11. Xuesong Yan, Hong Yao, Qingzhong Liang, Chengyu Hu, Yuanyuan Fan, Wenzhi Huang. Engineering Optimization Algorithm based on Particle Swarm Optimization. International Journal of Advancements in Computing Technology, 2013, Vol. 5, No. 3, pp. 113-121.
12. Xuesong Yan, Chengyu Hu, Hong Yao, Yuanyuan Fan, Qingzhong Liang, Chao Liu. Electronic Circuit Optimization Design Algorithm based on Particle Swarm Optimization. Przeglad Elektrotechniczny, 2013, Vol.89, Issue 03b, pp. 50-52.
13. Xuesong Yan, Qinghua Wu, Hammin Liu. Orthogonal Particle Swarm Optimization Algorithm and Its Application in Circuit Design. TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013, vol. 11, No. 6, pp. 1948-1955.
14. Wu, Qinghua, Liu, Hanmin, Yan, Xuesong. An improved design optimisation algorithm based on swarm intelligence. International Journal of Computing Science and Mathematics, 2014, v 5, n 1, pp. 27-36.
15. X.S. Yan, Kang Wang. Designing Electronic Circuits Using Cultural Algorithms. Proceedings of Third International Workshop on Advanced Computational Intelligence, 2010, pp. 299-303.
16. X.S. Yan, Qinghua Wu. Electronic Circuits Optimization Design Based On Cultural Algorithms International Journal of Information Processing and Management, 2011, Vol.2(1), pp. 49-56.
17. Xuesong Yan, Wei Chen, Qinghua Wu, Hanmin Liu. Research of Embedded Hardware Optimization Design Algorithm. International Journal of Computer Science Issues, 2012, Vol. 9, Issue 6, No. 2, pp. 70-78.
18. Xuesong Yan, Chengyu Hu, Hong Yao, Qingzhong Liang, Yuanyuan Fan. Circuit Optimization Design based on Improved Cultural Algorithm. International Journal of Advancements in Computing Technology, 2013, Vol. 5, No. 3, pp. 122-130.
19. Xuesong Yan, Qinghua Wu, Chengyu Hu, Hong Yao and Hanmin Liu. Electronic Circuit Optimization Design Algorithm based on Adaptive Culture Search. International Journal of Control and Automation, 2013, Volume 6, No. 4, pp. 11-28.
20. Yan, Xuesong, Hu Chengyu, Yao Hong, Wu Qinghua, Liu Hanmin, Adaptive Cultural Algorithm for Combinational Digital Circuit Sensor Design. Sensor Letters. 2015, Volume 13, Number 2, pp. 127-129.
21. X.S. Yan, Q.H. Wu and H.M. Liu. Orthogonal Evolutionary Algorithm and its Application in Circuit Design. Przeglad Elektrotechniczny, 2012, Vol. 88, Issue 05b, pp. 7-10.
22. Xue Song Yan, Qing Hua Wu, Cheng Yu Hu, Qing Zhong Liang. Circuit Optimization Design Using Evolutionary Algorithms. Advanced Materials Research, 2011, Vol.187, pp. 303-308.
23. Xuesong Yan, Hong Yao, Qingzhong Liang,Chengyu Hu. Research of Digital Circuit Optimization Design Algorithm. Advances in Information Sciences and Service Sciences, 2012, Vol. 4, No. 21, pp. 556-563.
24. Yan, X., Wu, Q., Liang, Q., Hu, C. and Yao, H. Research of electronic circuit optimisation design algorithm. Int. J. Computer Applications in Technology, 2013, Vol. 46, No. 3, pp. 267–275.
25. Goldbeg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning, Reading, Mass.: Addison-Wesley, 1989.
26. Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs, 3rd Ed. Berlin: Springer – Verlag, 1996.
27. R. Bellman. Dynamic Programming, Princeton University Press, 1957.
28. D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1997, 1(1), pp. 67-82.
29. R. Reynoids. An introduction to cultural algorithms. Proceedings of the 3rd Annual Confer-

ence on Evolutionary Programming, Singapore, 1994, pp. 131-139.

30. CHUNG C. Knowledge-based approaches to self-adaptation in cultural algorithms. Ph.D. Thesis, Wayne State University, Detroit, Michigan, 1997.

31. Miller J.F., Job D., Vassilev V.K. Principles in the Evolutionary Design of Digital Circuits - Part I. Genetic Programming and Evolvable Machines, 2000, 1, pp. 8-35.

Research on the Modeling and Optimization on Wireless Ad-hoc Sensor Network for Natural Disease

Jinguo Zhao, Chen Min, Qingyun Luo

School of Computer Science and Technology, Hunan Institute of Technology, Hengyang, China

Corresponding author is Chen Min

Abstract

In this paper, the author studied the modeling and optimization on Wireless Ad-hoc sensor network for natural disease. The paper addresses the power consumption in wireless ad-hoc and sensor networks. It primary identifies the wireless aspects. It then shows by mathematical analysis the energy saving, improvement error probability and confirms the analysis by simulation results. After that, it considers the multipath fading problem by modeling, mathematical analysis for fading channels, in addition to it discusses how to mitigate the multipath fading by modulation and diversity, and verifies that by simulation.

Keywords: MODELING, OPTIMIZAITON, WIRELESS AD-HOC, SENSOR NETWORK, NATURAL DISEASE

1. Introduction

Modern technological progresses in sensor networks embedded computing technologies and wireless communications have enabled the design of lightweight low power intelligent monitoring devices. A key technical challenge in most wireless sensor networks is how to successfully arrange and manage the sensors and how to control data collection in order to get a minimum continuous delay, a maximum monitoring duration and a desired excellence of monitoring. The maximum monitoring duration is period of time that monitoring service is provided while excellence of monitoring captures possible performance and accuracy of data.

Modern technological progresses in sensor networks embedded computing technologies and wireless communications have enabled the design of lightweight low power intelligent monitoring devic-