Research of Embedded Hardware Optimization Design Algorithm

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

[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, China

Abstract

Embedded hardware design is important in real world applications, but with the increase of the hardware scale the traditional methods can not design them well. Cultural Algorithms are a class of computational models derived from observing the cultural evolution process in nature. Aiming at the disadvantages of basic Cultural Algorithms like being trapped easily into a local optimum, this paper improves the basic Cultural Algorithms and proposes a new algorithm to solve the overcomes of the basic Cultural Algorithms. The new algorithm keeps not only the fast convergence speed characteristic of basic Cultural Algorithms, but effectively improves the capability of global searching as well. We use our algorithm in circuit optimization design and relay volume optimization, the experiment results show our algorithm is effective for these problems.

Keywords: *Cultural Algorithm, Optimization Design, Circuit Design, Evolutionary Computation.*

1. Introduction

An important area in current research is the development and application of search techniques based upon the principles of natural evolution. Evolution can be viewed as s change in the genetic composition of a population of individuals over time. In a simplified form, evolution is result of the successive processes of reproduction and genetic variation followed by natural selection, which allows the fittest individuals to survive and reproduce, thus propagating their genetic material to future generations. We shall use this as a starting point in introducing evolutionary computation. The theory of natural selection proposes that the plants and animals that exist today are the result of millions of years of adaptation to the demands of the environment. At any given time, a number of different organisms may co-exist and compete for the same resources in an ecosystem. The organisms that are most capable of acquiring resources and successfully procreating are the ones whose descendants will tend to be numerous in the future. Organisms that are less capable, for whatever reason, will tend to have few or no descendants in the future. The former are said to be more fit than the latter, and the distinguishing characteristics that caused the former to be more fitness are said to be selected for over the characteristics of the latter. Over time, the entire population of the ecosystem is said to evolve to contain organisms that, on average, are more fit than those of previous generations of the population because they exhibit more of those characteristics that tend to promote survival.

Evolutionary computation techniques abstract these evolutionary principles into algorithms that may be used to search for optimal solutions to a problem. In a search algorithm, a number of possible solutions to a problem are available and the task is to find the best solution possible in a fixed amount of time. For a search space with only a small number of possible solutions, all the solutions can be examined in a reasonable amount of time and the optimal one found. This exhaustive search, however, quickly becomes impractical as the search space grows in size. Traditional search algorithms randomly sample or heuristically sample the search space one solution at a time in the hopes of finding the optimal solution. The key aspect distinguishing an evolutionary search algorithm from such traditional algorithms is that it is population-based. Through the adaptation of successive generations of a large number of individuals, an evolutionary algorithm performs an efficient directed search.

Evolutionary search is generally better than random search and is not susceptible to the hill-climbing behaviors of gradient based search.

Evolutionary computation methods such as Genetic Algorithms, Evolution Strategy, Genetic Programming, and Evolutionary Programming are very powerful tools in solving search and optimization problems such as NP-complete problems. Evolutionary algorithms have been successful in very diverse search and optimization problems due to their unbiased nature, which allows them to perform very well in situations with Little domain knowledge[1-2]. However, research has shown that considerable improvement in their performance can be achieved when problem-specific knowledge is used to adapt the problem solving process in order to identify patterns in the performance environment. The knowledge concerning these patterns in the problem solving experiences of the evolutionary population is used to influence the generation of candidate solutions to either promote more instances of desirable candidates or to reduce the number of less desirable candidates in the population.

In human societies, culture can be viewed as a vehicle for the storage of information that is potentially accessible to all members of the society, and that can be useful in guiding their problem solving activities. As such, groups that are able to support a cultural tradition can use their cultural heritage as a mechanism to bias the generation of individuals on at least a phenotypic level. This is done by facilitating the production of phenotypes that are promising in a given environment on the one hand, and deterring the production of phenotypes that are less likely to be productive on the other. Cultural Algorithms have been developed by Reynolds in order to model the evolution of the cultural component of an evolutionary computation system over time as it accumulates experience in solving a given set of problems [3]. As a result, Cultural Algorithms are likely to provide a useful framework within which to model self-adaptation in an evolutionary computation system. This paper improves the disadvantages of basic Cultural Algorithms being easily trapped into a local optimum and presents a new algorithm which proves to be more simply conducted and with more efficient global searching capability.

2. Cultural Algorithm

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. 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 pseudo-code description of the Cultural Algorithms is described as follows:

```
Begin
t=0;
Initialize population P(t);
Initialize belief space B(t);
Repeat
Evaluate P(t) ;
Update(B(t), accept(P(t))) ;
Generate (P(t), influence(B(t)) ;
t+=1 ;
Select P(t) from P(t-1) ;
Until (termination condition)
```

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.

In the basic cultural algorithms, the belief space uses the {S,N} structure represented[4]. The formal syntax for the belief space B , used in this study is: $B = S\left|N\right|\langle S,N\rangle$, where S denotes structure for situational knowledge and N denotes structures for normative knowledge. The definition above means the belief space can consist of situational knowledge only, normative knowledge only, or both. The situational knowledge S is represented formally as a pair wise structure: $S = \langle < E_1, E_2, ..., E_e >, adjust_E(e) \rangle$, where E_i represent an ith best exemplar individual in the evolution history. There can be e best exemplars in S as s set that constitutes the situational knowledge. Each exemplar individual has n parameters and a performance value. $adjust_E(e)$ is the belief space operator applied to update e number of exemplar individuals in S. The normative knowledge N, a set of interval information for each of the n parameters is defined formally as 4-tuple: $N = \langle I_j, L_j, U_j, adjust_N \rangle, j = 1,2,...,n$, where I_j denotes the closed interval of variable j, that is a continuous set of real numbers x represented as a ordered number pair: $I_j = [l_j, u_j] = \{x \mid l_j \le x \le u_j, x \in R\}$. l_j (lower bound) and u_j (upper bound) are initialized by the give domain values. L_j represents the performance score of the lower bound l_j for parameter j. U_j represents the performance score of the upper bound u_j for parameter j.

For the update function, we defined like this: $S = \{st\}$, select the best individual st update the situation knowledge S in belief space. The update process follows the Eq. (1):

$$s^{t+1} = \begin{cases} x_{best}^t & f(x_{best}^t) < f(s^t) \\ s^t & \text{others} \end{cases} \quad (1)$$

where x_{best}^t denotes tth best individual.

Update the normative knowledge N in belief space uses the Eq. (2):

$$l_i^{t+1} = \begin{cases} x_{j,i} & x_{j,i} \le l_i^t \text{ or } f(x_j) < L_i^t \\ l_i^t & \text{others} \end{cases}$$

$$L_i^{t+1} = \begin{cases} f(x_j) & x_{j,i} \le l_i^t \text{ or } f(x_j) < L_i^t \\ L_i^t & \text{others} \end{cases}$$

$$u_i^{t+1} = \begin{cases} x_{j,i} & x_{j,i} \ge u_i^t \text{ or } f(x_j) < U_i^t \\ u_i^t & \text{others} \end{cases}$$

$$U_i^{t+1} = \begin{cases} f(x_j) & x_{j,i} \ge u_i^t \text{ or } f(x_j) < U_i^t \\ U_i^t & \text{others} \end{cases}$$

(2)

In our CA, the knowledge represented in the belief space can be explicitly used to influence the creation of the offspring via an influence function. In our sliding window model, the strategy can be simply described as follows. The first is if a parent is in a promising region, the offspring are created by randomly changing the problem parameters of the parent just a little. In this case, the normative knowledge applies. The offspring $x_{j,i}^{t+1}$, will be created by using this normative knowledge as a beacon to attract the parent $x_{j,i}^t$ to move a copy toward the current sliding window, the influence function defined by the Eq. (3).

$$x_{j,i}^{t+1} = \begin{cases} x_{j,i}^t + |size(I_i) * N(0,1)| & x_{j,i}^t < l_i^t \\ x_{j,i}^t - |size(I_i) * N(0,1)| & x_{j,i}^t > u_i^t \\ x_{j,i}^t + \lambda_1 * size(I_i) * N(0,1) & others \end{cases} \quad (3)$$

The second is if a parent is in an unpromising region, moving a copy of the parent to a more promising region can be used to create a new offspring. In this case, the constraint knowledge applies. The creation of offspring will be affected by the characteristic of the cells within the sliding window, the influence function defined by the Eq. (4).

$$x_{j,i}^{t+1} = \begin{cases} x_{j,i}^t + |size(I_i) * N(0,1)| & x_{j,i}^t < S_i^t \\ x_{j,i}^t - |size(I_i) * N(0,1)| & x_{j,i}^t > S_i^t \\ x_{j,i}^t + \lambda_1 * size(I_i) * N(0,1) & others \end{cases} \quad (4)$$

In order to verify the validity of the cultural algorithm, we using four benchmarks function to verify the algorithm's effectiveness.

F1: Schaffer function

$$\min f(x_i) = 0.5 - \frac{(\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5)}{[1 + 0.001(x_1^2 + x_2^2)]^2}, -10 \le x_i \le 10$$

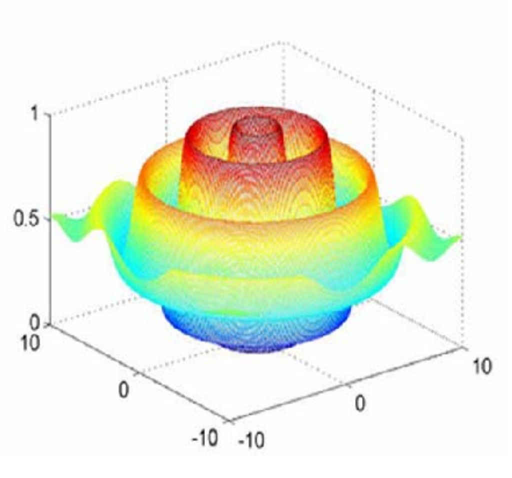

Fig. 1 Schaffer function

In this function the biggest point is that the overall situation (0, 0), the largest in the overall points for the center, to 3.14 for the radius of a circle on the overall situation from numerous major points of the uplift, and, This function has a strong shock, therefore, it is difficult to find a general method of its global optimal solution.

F2: Shubert function

$$\min f(x,y)=\left\{\sum_{i=1}^{5} i\cos\left[(i+1)x+i\right]\right\}\times\left\{\sum_{i=1}^{5} i\cos\left[(i+1)y+i\right]\right\},$$
$$x,y\in[-10,10]$$

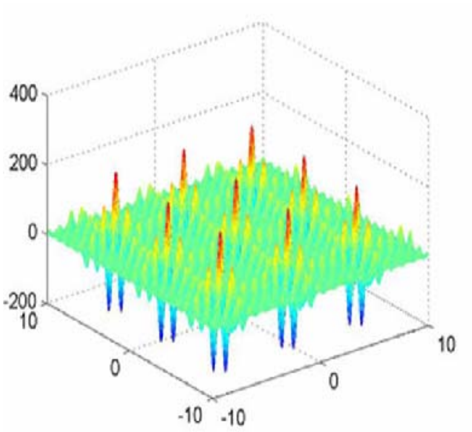

Fig. 2 Shubert function

This function has 760 local minimum and 18 global minimum, the global minimum value is -186.7309.

F3: Hansen function

$$\min f(x,y)=\sum_{i=1}^{5} i\cos((i-1)x+i)\sum_{j=1}^{5} j\cos((j+1)y+j),$$
$$x,y\in[-10,10]$$

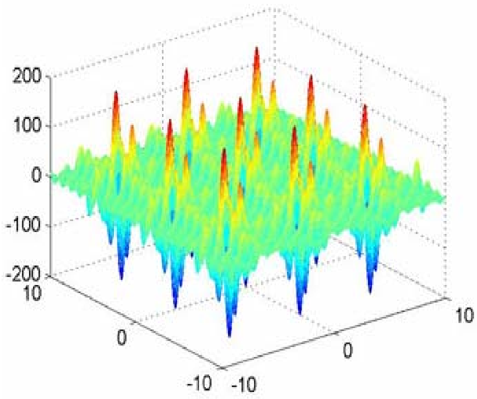

Fig. 3 Hansen function

This function has a global minimum value -176.541793, in the following nine point (-7.589893, -7.708314)、(-7.589893, -1.425128)、(-7.589893, 4.858057)、(-1.306708, -7.708314)、(-1.306708, -1.425128)、(-1.306708, 4.858057)、(4.976478, -7.708314)、(4.976478, -7.708314)、(4.976478, 4.858057) can get this global minimum value, the function has 760 local minimum.

F4: Camel function

$$\min f(x,y)=\left(4-2.1x^2+\frac{x^4}{3}\right)x^2+xy+\left(-4+4y^2\right)y^2,$$
$$x,y\in\left[-100,100\right]$$

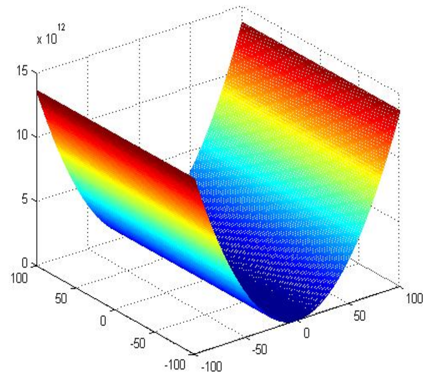

Fig. 4 Camel function

Camel function has 6 local minimum (1.607105, 0.568651)、(-1.607105, -0.568651)、(1.703607, -0.796084)、(-1.703607, 0.796084)、(-0.0898,0.7126)和(0.0898,-0.7126), the (-0.0898,0.7126) and (0.0898,-0.7126) are the two global minimums, the value is -1.031628.

We run cultural algorithm and get the results described in the Fig. 5.

FUNCTION	GENERATIONGS	CONVERGENCE RATE	OPTIMAL
F1	19	100%	1.000000
F2	10	100%	-186.730909
F3	12	100%	-176．541793
F4	42	56%	-1.031628

Fig. 5 Experimental results for the four Benchmarks

3. Embedded Hardware Optimization Design

3.1 Circuit Optimization Design

Digital circuits are widely used in many areas of irreplaceable role. Small-scale digital circuits can be designed manually by artificial means. With the size of digital circuits' increases, the difficulty of manual design methods increases exponentially. When the size reaches a certain level, manual design method is infeasible. In this paper, we use cultural algorithm to solve this problem. We use this algorithm design two-bit full adder circuit (only use six-gates) and compared the result with manual method (use ten-gates).

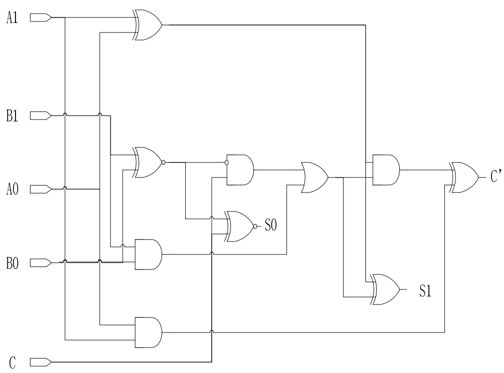

Fig. 6 Two-bit full adder's circuit design by manual

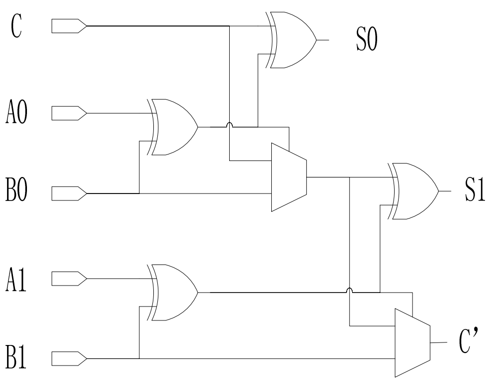

Fig. 7 Two-bit full adder's circuit design by CA

We also use this algorithm design Parity Checker's circuit. The following figures are the results (for the eight-bit, nine-bit, ten-bit, eleven-bit and twelve-bit Parity Checker) and Table 1 is the statistics of experiment results.

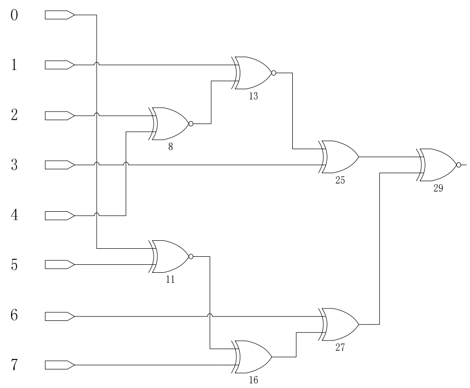

Fig. 8 Eight-bit even checker's circuit design by CA

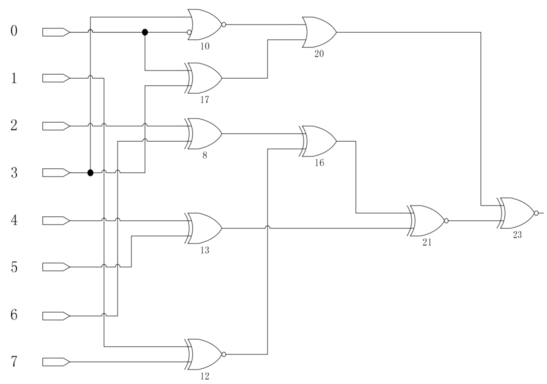

Fig. 9 Eight-bit odd checker's circuit design by CA

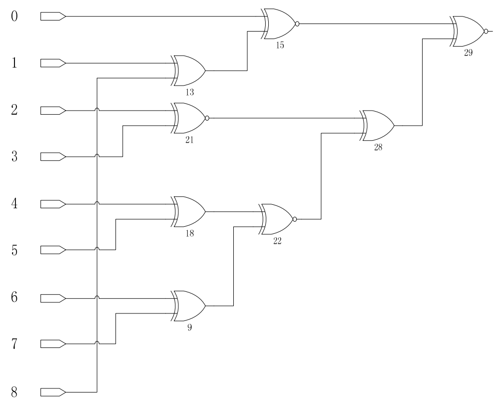

Fig. 10 Nine-bit even checker's circuit design by CA

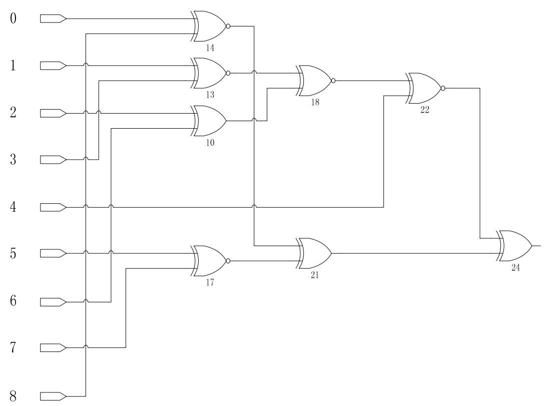

Fig. 11 Nine-bit odd checker's circuit design by CA

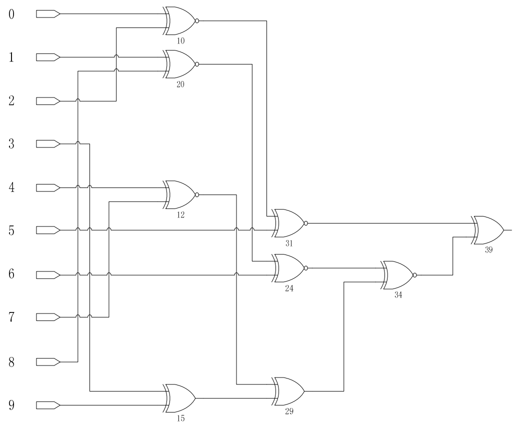

Fig. 12 Ten-bit even checker's circuit design by CA

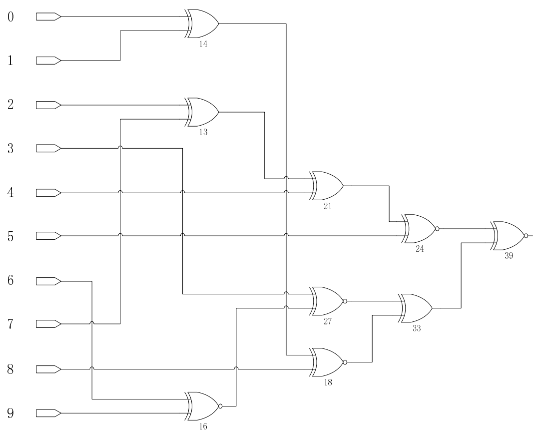

Fig. 13 Ten-bit odd checker's circuit design by CA

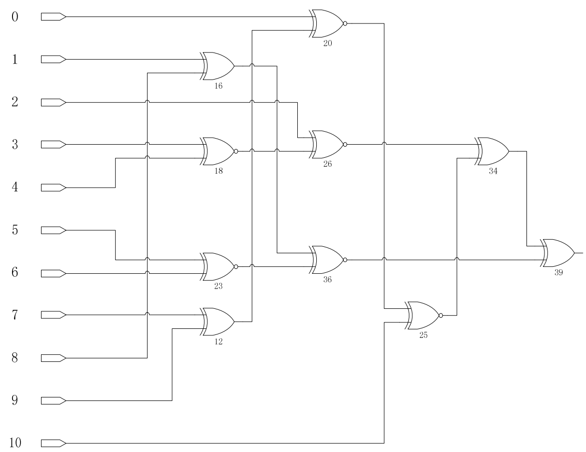

Fig. 14 Eleven-bit even checker's circuit design by CA

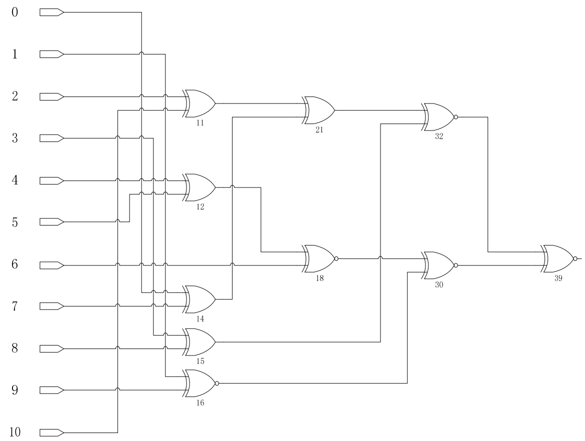

Fig. 15 Eleven-bit odd checker's circuit design by CA

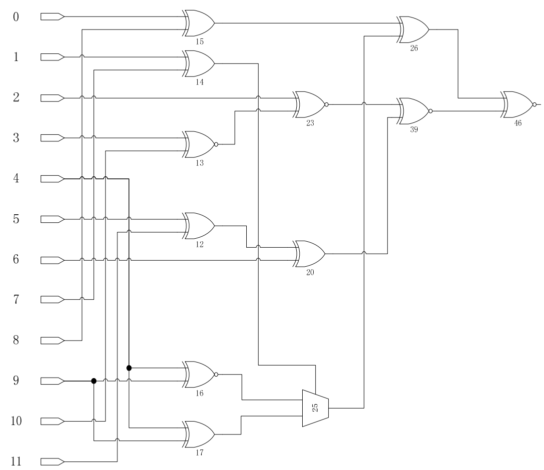

Fig. 16 Twelve-bit even checker's circuit design by CA

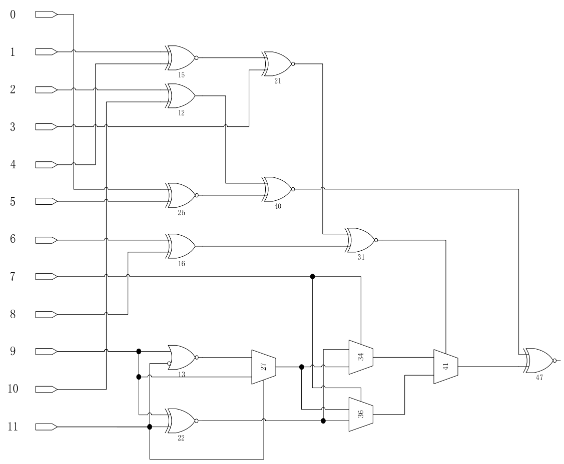

Fig. 17 Twelve-bit odd checker's circuit design by CA

Table 1: Statistics of experiment results

Parity Checker's Circuit	***Design Time***	***Test Number***	***Success Rate***	***Optimal Gates***
Eight-bit Even	30(s)	10	100%	7
Eight-bit Odd	30(s)	10	100%	7
Nine-bit Even	50(s)	10	90%	8
Nine-bit Odd	50(s)	10	90%	8
Ten-bit Even	2(m)	10	90%	9
Ten-bit Odd	2(m)	10	90%	9
Eleven-bit Even	5(h)	10	80%	10
Eleven-bit Odd	5(h)	10	80%	10
Twelve-bit Even	12(h)	10	80%	11
Twelve-bit Odd	12(h)	10	80%	11

3.2 Relay Optimization Design

Optimization design of relay products determine the design parameters in the given load conditions or environmental conditions, the state of relay products, geometry or other factors within the scope of restrictions and make sure of the design parameters, object function, constraints in order to form an optimized design model, and select the appropriate optimization method to obtain the best design of a series of work. Mathematical model of the relay volume involves in mechanical, electrical, magnetic, thermal, etc., the objective function and constraints are highly nonlinear function and the traditional optimization algorithm trapped into the local minimum easily. In this paper, we use cultural algorithm and get more optimal result.

The relay optimization goal is to ensure that electrical and reliable action and release the premise, and strive to save energy, materials, and reduce Core collision energy, and other factors, to prevent contact bounce, so it is the constrained nonlinear programming problem. We can describe the relay volume's mathematic model as equation (5) [5]:

$$\min V(x) = \frac{1.5913*10^{-10}}{x_2^3}\left[x_3 x_1^2 + x_3 + 4x_1 + \frac{1}{2}\left(x_4^2 - \frac{1}{x_4^2}\right)\right]$$

$$St.$$

$$g_1(X) = 1.0764*10^{-6} * \frac{x_3^2 x_1 (x_1 - 1)}{x_2^3 (x_1 + 1)}\left[\frac{3.1416 x_4^2}{x_2^3}\left(1 + \frac{x_4^2}{4x_1^2}\right) + \frac{32(x_4 - x_4^{-1})}{(1 - x_4^{-2} + 4x_2)^2} - 0.9\right] - 1.8 \geq 0,$$

$$g_2(X) = 1.4 - 1.4112*10^{-6} * x_3 * \sqrt{\frac{(x_1 - 1)x_1}{(x_1 + 1)x_2}} *$$

$$\left\{\frac{3.1416}{x^2}\left(x_4^2 + \frac{x_4^4}{4x_1^2}\right) + 0.9x_2 + \frac{3x_4^3 + x_4^2 - 1}{x_4^2} + \frac{8x_4(x_4^2 - 1)}{x_4^2 + 4x_2 x_4^2 - 1} + \frac{3.1416 x_3}{\ln\left(x_1 + \sqrt{x_1^2 - 1}\right)}\right\} \geq 0,$$

$$g_3(X) = 85 - 0.925 * \frac{(x_1^2 - 1)}{x_1 x_2} \geq 0,$$

$$g_4(X) = x_1 - x_4 > 0,$$

$$g_5(X) = \frac{x_3}{x_1} - 1.33333 \leq 0,$$

$$g_6(X) = 1.5 - \frac{x_3}{x_4} \geq 0 \qquad (5)$$

In this equation, variable x1's value calculated by the design parameter Coil outer radius, x2's value calculated by the design parameter Core-radius, x3's value calculated by the design parameter Coil height and x4's value calculated by the design parameter Pole radius, so the optimization goal is to get the optimal value of the four variables.

In [5] has give a example of the four parameters of a relay. In Table 2 shows the four variable's value before optimization, Table 3 shows the optimized value for the four variables in [5] and Table 4 shows the four variable's value design by CA. When we get these design variables, we can use equation (5) to calculate the relay's volume, the results are show in Table 5.

Table 2: Design parameter before optimization

Design Parameter	***Value(mm)***	***Design Variable***
Coil outer radius	6.00	$x_1 = R_0 / r_c = 2.60870$
Core-radius	2.30	$x_2 = \delta / r_c = 0.16087$
Coil height	9.00	$x_3 = h_c / r_c = 3.91304$
Pole radius	4.00	$x_4 = r_j / r_c = 1.73913$

Table 3: Optimized design parameter

Design Parameter	***Value(mm)***	***Design Variable***

Coil outer radius	5.26	$x_1 = {R_0}/{r_c} = 2.21259$
Core-radius	2.38	$x_2 = {\delta}/{r_c} = 0.15571$
Coil height	7.71	$x_3 = {h_c}/{r_c} = 3.24290$
Pole radius	4.98	$x_4 = {r_j}/{r_c} = 2.09468$

Table 4: Optimized design parameter by CA

Design Parameter	***Value(mm)***	***Design Variable***
Coil outer radius	5.063	$x_1 = {R_0}/{r_c} = 2.13375$
Core-radius	2.373	$x_2 = {\delta}/{r_c} = 0.155918$
Coil height	7.585	$x_3 = {h_c}/{r_c} = 3.19639$
Pole radius	5.063	$x_4 = {r_j}/{r_c} = 2.13368$

It can be seen from Table 5, the cultural algorithm to optimize the mathematical model of the relay volume than before the optimization, the volume reduces 26.05%, compared with the optimization method proposed in [5], the volume is reduced by 5.7%.

Table 5: Results Comparison

Relay Volume Optimization Use CA(mm^3)	***Relay Volume Optimization in [5] (mm^3)***	***No Optimization(mm^3)***
1194.41	1266.6	1617.78

4. Conclusions

This paper introduces a new algorithm-Cultural Algorithm in embedded hardware optimization design. CA is 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. In order to solve the problems in real world, we design the cultural algorithm follow the idea of basic CA and test the algorithm's effective use the benchmark functions.

Circuit optimization design and Relay optimization design are all the challenge problems, traditional method can not deal with them very well, We are design the cultural algorithm for the two problems, and use the algorithm design the circuit and relay volume optimization design, the experiment results show this method is effectiveness.

Acknowledgments

This paper is supported by the Provincial Natural Science Foundation of Hubei (No. 2011CDB334 and 2011CDB346), the Fundamental Research Founds for National University, China University of Geosciences (Wuhan) and National Civil Aerospace Pre-research Project of China.

References

[1] D. Fogel, L. Fogel, and J. W. Atmar, "Meta-evolutionary programming," In Proc. of the 24h Asilornar Conf on Signals, Systems, and Computers, edited by R.R. Chen. Pacific Grove, CA: IEEE Comp. Soc. Press, 1995, pp. 540-545.

[2] R. Reynolds and C. Chung, "Knowledge-based self-adaptation in evolutionary programming using cultural algorithms," in Proceedings of LEEE International Conference on Evolutionary Computation (ICEC '97), 1997.

[3] R. Reynoids, "An introduction to cultural algorithms," in Proceedings of the 3rd Annual Conference on Evolutionary Programming, Sebald, AX; Fogel, L.J. (Editors), River Edge, NJ, World Scientific Publishing, 1994, pp. 13 1-139.

[4] CHUNG C, "Knowledge-based approaches to self-adaptation in cultural algorithms", Ph.D. Thesis, Wayne State University, Detroit, Michigan, USA, May 1997.

[5] Lingling Li, "Construction Method of Relay product design and evaluation of expert system", Master Thesis, Hebe University of Technology, Shijiazhuang, China, 2004.

[6] X.S. Yan, Q.H. Wu, "Function Optimization Based on Cultural Algorithms", Journal of Computer and Information Technology, Vol.2, 2012, pp.152-158

[7] X.S. Yan et,al, "Electronic Circuit Automatic Design Based on Genetic Algorithms", Procedia Engineering, Vol.15, 2011, pp.2948-2954.

[8] X.S. Yan, Q.H. W et.al, "Electronic Circuits Optimization Design Based On Cultural Algorithms", International Journal of Information Processing and Management. 2(1), 2011, pp. 49-56.

[9] X.S. Yan et.al; "Designing Electronic Circuits Using Cultural Algorithms", Proceedings of Third International Workshop on Advanced Computational Intelligence, 2010, pp. 299-303.

[10]Xuesong Yan, Qinghua Wu, et.al, "An Efficient Function Optimization Algorithm based on Culture Evolution",

International Journal of Computer Science Issues, Vol. 9, No. 2, 2012, pp. 11-18.
[11]X.S. Yan, Q.H. Wu and H.M. Liu, "Orthogonal Evolutionary Algorithm and its Application in Circuit Design", Przeglad Elektrotechniczny (Electrical Review), Vol.88, Issue 05b, 2012, pp.7-10.
[12]X.S. Yan et, al, "Orthogonal evolutionary algorithm and its application in relay volume optimization design", Computer Engineering and Applications, 47(18), 2011, pp.215-217. (in Chinese).
[13]Xue Song Yan, Qing Hua Wu, Cheng Yu Hu, Qing Zhong Liang, "Circuit Design Based on Particle Swarm Optimization Algorithms", Key Engineering Materials, Vols. 474-476, 2011, pp.1093-1098.
[14]Xue Song Yan, Qing Hua Wu, Cheng Yu Hu, Qing Zhong Liang, "Circuit Optimization Design Using Evolutionary Algorithms", Advanced Materials Research, Vol.187, 2011, pp.303-308.
[15]Xue Song Yan, Qing Hua Wu, Cheng Yu Hu, Qing Zhong Liang, "Research of Space Electronic Circuit Optimization Design", Applied Mechanics and Materials, Vols.48-49, 2011, pp.932-936.
[16]X.S. Yan, Qing Hua Wu et.al, "Design Electronic Circuits Using Evolutionary Algorithms", Journal of Next Generation Information Technology, 1(1), 2010, pp.127-139.
[17]X.S. Yan et, al, "Electronic Circuits Automatic Design Algorithm", Proceedings of Sixth International Conference on Natural Computation, 2010, pp.2334-2337.
[18]X.S. Yan et, al, "Application of Cultural Algorithm in Stock Data Modeling", Computer Engineering & Science, Vol. 32, No.6, 2010, pp. 68-71.(in Chinese).

Xuesong Yan associate professor received him B.E. degree in Computer Science and Technology in 2000 and M.E. degree in Computer Application from China University of Geosciences in 2003, received he Ph.D. degree in Computer Software and Theory from Wuhan University in 2006. He is currently with School of Computer Science, China University of Geosciences, Wuhan, China and now as a visiting scholar with Department of Computer Science, University of Central Arkansas, Conway, USA. He research interests include evolutionary computation, data mining and computer application.

Wei Chen received him B.E. degree in Computer Science and Technology in 2012. He is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Qinghua Wu lecturer received her B.E. degree in Computer Science and Technology in 2000, M.E. degree in Computer Application in 2003 and Ph.D. degree in Earth Exploration and Information Technology Theory from China University of Geosciences in 2011. She is currently with School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan, China. Her research interests include evolutionary computation, image processing and computer application.

Hanmin Liu associate professor. He is currently as a Ph.D candidate of School of Computer Science, China University of Geosciences, Wuhan, China. He research interests include evolutionary computation and applications.