An Improved Genetic Algorithm Based On Stages Hybridization For Evolvable Hardware Design

Huicong Wu[1,a], Jinze Wang[2,b]

[1]College of information science and engineering, Hebei University Of Science and Technology,Shijiazhuang,Hebei 050018,China

[2]College of information science and engineering, Hebei University Of Science and Technology,Shijiazhuang,Hebei 050018,China

[a]whc@hebust.edu.cn, [b]kdwangjinze@163.com

Keywords: circuit design; evolvable hardware; genetic algorithm; stages hybridization

Abstract: Facing the complexity and variability of electronic circuit design, evolvable hardware has gradually become one of the effective methods to design hardware. Algorithm influences the direction and progress of evolution. In order to improve algorithm and solve the current problem that the evolutionary time is long, the convergence speed is slow and the computation is large. This paper proposed an improved stages hybridization genetic algorithm. Using evolvable hardware platform, improved algorithm adopted the stages strategy and simulated the natural population hybridization based on genetic principle. The results show that the improved algorithm has better evolution results than the standard algorithm, whose convergent speed is faster and evolutionary cycle is shorter. Offspring circuits have a higher adaptability. Improved algorithm reduces calculation, and stages parallel strategy shortens time consuming and saves resources. Finally the optimal values of grouping number N and evolutionary generation T are found to improve the algorithm accuracy, which is beneficial to the whole evolution.

Introduction

Evolvable hardware is a new adaptive hardware system, which combines evolutionary algorithm and reconfigurable devices[1]. It implements a global search by using genetic algorithm, uses field reconfigurable device as carrier, and defines fitness function as an evaluation method to simulate the biological evolution algorithm[2]. In the premise of not relying on priori knowledge, the structure of the circuit is automatically adjusted to adapt to hardware application environment changes, and finally it meets the requirements of hardware design and gets the optimal target circuit. Research on the evolvable hardware is very important in space, deep sea exploration and application of nuclear industry[3,4]. In recent years, it has become a hot research topic in China and foreign countries.

Improved Algorithm Design

Evolvable hardware design is divided into analog circuit evolution design and digital circuit evolution design[5]. This paper was based on the digital circuit evolution design platform, and proposed an improved stage hybrid genetic algorithm as the platform genetic algorithm. The results show that this improved genetic algorithm can improve the diversity of population, shorten the cycle of evolution, reduce the amount of evolutionary computation, and generate the target circuit which more meets the design requirements.

Stage hybrid genetic algorithm initializes population and the initial population is divided into N sub populations, each sub population separately evolve T generations, then excellent individuals of each population will be replaced by each other, and a certain proportion of adverse individuals will be abandoned. N recomposition sub populations continue to evolve T generations, and so repeatedly the T generation offsprings have the advantages of hybridization. Finally, the design circuit which meets the design requirements is gotten.

The genetic algorithm operators are the core part of the algorithm, which often determine the diversity of the population and search direction[6]. It mainly includes the cross operator, mutation operator and selection operator[7,8]. In this paper, mutation operator and selection operator are only used to evolve population. The mutation operator is used in all individual gene of N sub populations, then decode gene and simulate circuit. According to evaluation function, the fitness of individual is sorted in each sub population, and the candidate individual enters the next generation, finally the sub offspring population is obtained by the T generation evolution shown in Fig. 1.

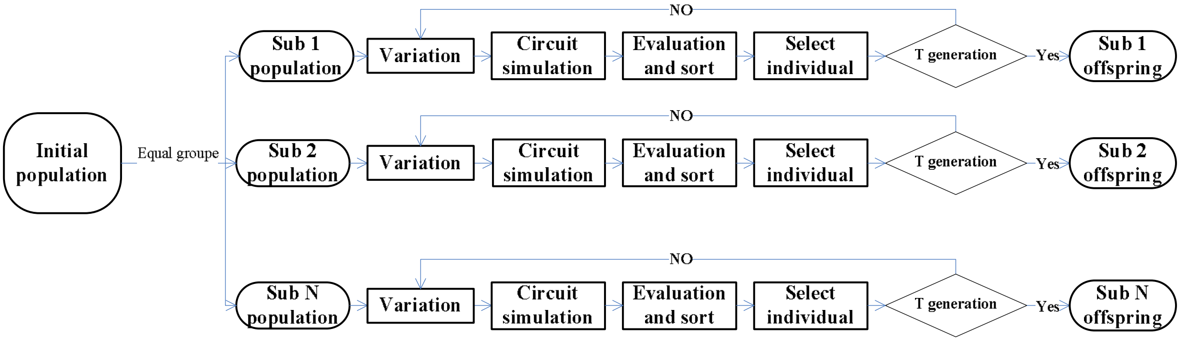

Fig. 1. Algorithm basic process

In each sub offspring population, a certain proportion excellent individuals were taken out and replaced by each other, while a certain proportion of adverse individuals will be abandoned. In order to make the equal number of sub offspring populations in the evolution process, excellent individuals were mutated to replenish the position of abandoned individuals. After repeating M times, the excellent circuit which more meet the design requirements is gotten shown in Fig. 2.

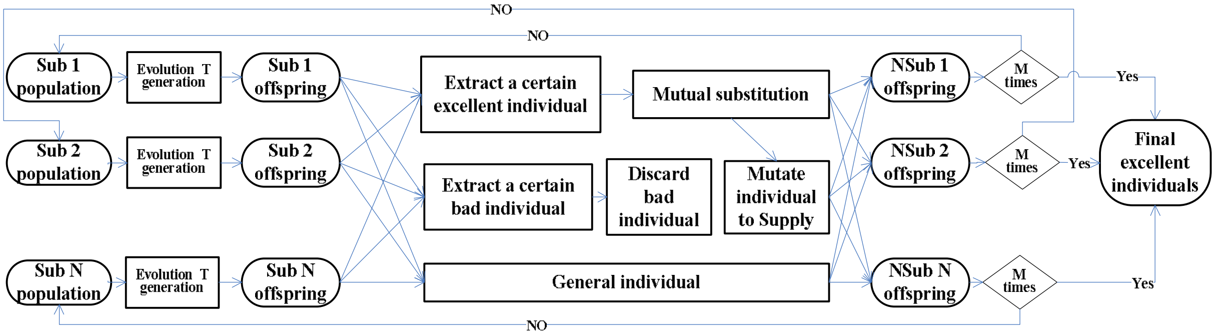

Fig. 2. Stage hybridization strategy

Digital Circuit model

The chromosome encoding way is divided into two kinds: indirect encoding way and direct encoding way[9]. Direct encoding way takes the configurable bit string of reconfigurable hardware as a chromosome, and the evolutionary results can be directly used for hardware configuration[10]. In digital circuit design, the logic gate is taken as the basic unit of encoding, which includes two inputs, one output and the gate category, such as Fig. 3 represents a two inputs one output And logic gate.

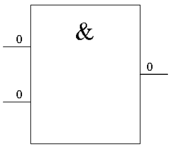

Fig. 3. And logic gate

Fig. 4 shows a model of digital circuit. It consists of a number of logic gates, from which is treated as a node, and the whole circuit model is composed of $r*s$ nodes (r rows and s columns). The model also has p inputs and q outputs. The design circuit is formed by the free combination of nodes, by which some functions are completed. Input of the first column nodes is from p inputs, after each input of the next column nodes is from the output of the previous column, by doing this, output is determined by the last column. If the number of model logic gates is predetermined, all logic gate genetic units will be connected to a gene string as a chromosome, which represents the entire circuit encoding. According to the coding way, the population can be initialized.

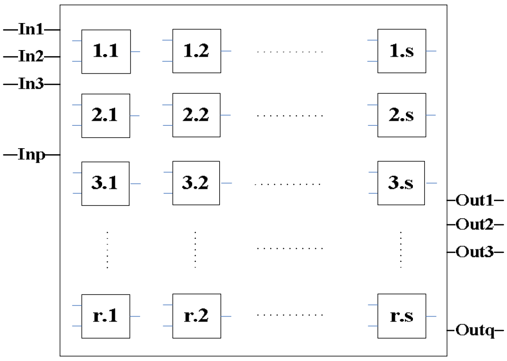

Fig. 4. Digital circuit model

In this paper, logic gate is used to design digital circuit. The logic gate inputs and gate category are regarded as the gene elements to encode directly. each logic gate input is represented respectively by e bits($e= log_2r$). The logic gate category is represented by *2* bits, and the simple categories of logic gate are shown in Table 1.

Table 1 Different category logic gates

Function umber	*Function*	*Symbolic representation*
00	and	*
01	or	+
10	not	-
11	empty	

In encoding the entire circuit, each node gene is represented by $2e+2$ bits. According to the column priority encoding way, the first column nodes are coded firstly, then the next column nodes are did and so on. Finally the whole circuit chromosome is composed of $r*s*$ $(2e+2)$ bits of $r*s$ nodes. Each individual of the population is represented by a chromosome.

The evaluation function is an important part of the evolution. The value is the embodiment of individual fitness. The higher the value is, the stronger the ability of the individual adapts, the more close the optimal target circuit is. In the digital circuit design, according to the truth table, the actual output value is compared with the expected output value. Eq. 1 is regarded as the evaluation of individual fitness function.

$$f(x)=\frac{1}{m}\sum_{x=1}^{m}(1-Qout\oplus Qobj)\ . \tag{1}$$

In the Eq. 1, m represents the the number of input and output combinations in the truth table. $Qout$ represents the actual output of per individual response. $Qobj$ represents the expected output of per individual response. $\oplus$ is xor symbol, if the two parameters are different, the result is 1, and if the two parameters are same, the result is 0.

The individual fitness $f\ (x)$ of each population is ordered, forming a vector Fi from high to low, which is shown in Eq. 2. According to the selection strategy, the next generation individuals are chosen and continue to evolve. n represents the number of individuals in the population, and i represents the identifier of sub populations.

$$F_i=\{f(1),f(2),...f(n)\}. \tag{2}$$

Mutation operator is used in the population evolution process, which changes the input parameter or category parameter of the node genes in each individual. Mutation operator improves the population diversity, expands the search scope, and finds a better circuit individual.

Experiments and Results

This paper uses the above model platform to simulate 2-3 binary multiplier. Circuit model uses 7*7 logic gate matrix, and a single individual encoding length is 392 (7*7* (2*3+2) =392) bits. Circuit model also has 5 input terminals and 5 output terminals, form which the input parameters are 2 bits and 3 bits. Corresponding to the truth table, There are 32 (2^{2+3}=32) kinds of combinations, so the *m* is 32 in Eq. 1. Improved genetic algorithm and standard genetic algorithm are carried on the population evolution experiments, and the detailed evolution parameters are shown in Table 2.

Table 2 Evolution parameters

Option	Improved algorithm	Standard algorithm
Sub population number N	4	1
Population size n	25	100
T	500	2000
M	4	0
Variation rate	10%	10%
Selection rate	30%	30%
Abandon rate	20%	0
e	3	3
*r*s*	7*7	7*7
p	5	5
q	5	5

The detailed evolution process is that direct encoding way initializes population; the initial population is mutated by genetic operator, then gene string is decoded and simulated to circuit; according to the truth table, the actual output is compared with the expected output, and using Eq. 1 calculates the fitness value of each individual, which is sorted from high to low; according to the choice strategy, candidates are chosen from the sort and put into the next generation evolution. The evolutionary process of improved algorithm increases the selection strategy: every 500 generations evolution, individuals among populations are selected, abandoned and restructured based on the improved algorithm, and then the recombinant sub offspring population enter the next cycle, until 4 times the evolution is terminative.

1) After evolution experiment repeating 100 times, the relationship between average fitness function *favg (x)* and evolution generation *x* is shown in Fig. 5.

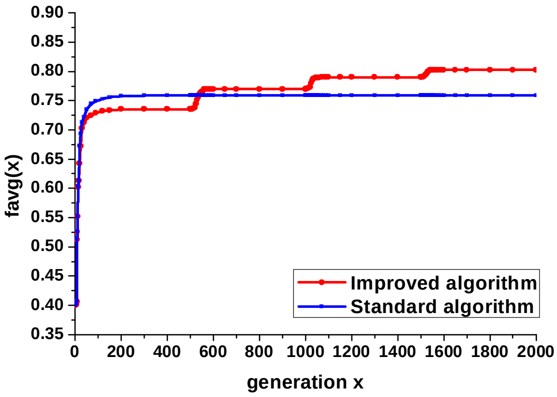

Fig. 5. The two algorithms' favg (x)

Fig. 5 shows that the population fitness has improved based on the improved genetic algorithm after each hybridization, of which each time the good genes of other populations is introduced, so the offspring has better adaptability. Although the magnitude of the increase is smaller, the final individual adaptation degree is higher, and more close to the target circuit individual. Compared with the standard algorithm, the convergence rate of the improved algorithm is faster. The hybrid strategy expands the search population scope and effectively prevents the occurrence of the local optimal solution. Compared with the standard algorithm, improved algorithm reduces the evolutionary period and improves the evolutionary efficiency.

2) In the evolution process, the calculation and time consuming of genetic operations are not big for two algorithms, however, the sorting comparison is main factor that affects the evolution time and calculation. The comparing experimental data is shown in Table 3.

Table 3 Sorting comparison of different algorithms

Option	Improved algorithm	Standard algorithm
Average comparison number	$(25^2/2)*4*500*4$	$(100^2/2)*2000$
Average time consuming	25*500*4	100*2000

Table 3 shows that the average time consuming of the improved algorithm is 2500000 times less than that of the standard algorithm. The improved algorithm efficiency increases 4 times in time consuming, which reduces the computation and saves the system resources. The average comparison number of the improved algorithm is 50000, which is a quarter of the standard algorithm comparison number. Improved algorithm reduces the evolutionary period, improves the evolutionary efficiency, and is beneficial to parallel computing.

3) This paper further studies the effect of group number N and evolutionary generation T on the algorithm result. Fig. 6 shows that the relationship of the average fitness function *favg (x)* and the group number N. Fig. 7 shows that the relationship of the average fitness function *favg (x)* and the evolutionary generation T.

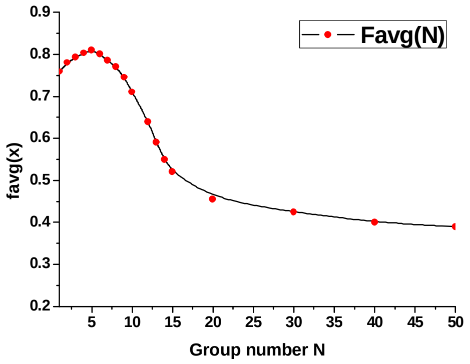

Fig. 6. The relationship of favg (x) and N

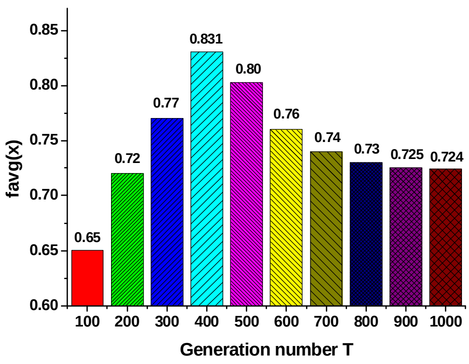

Fig. 7. The relationship of favg (x) and T

Fig. 6 shows that when the number of divided group is 1, it is the standard algorithm evolution. When the group number N initially increases, because many groups evolve synchronously, expanding the search area, sub populations of hybrid strategy improve the diversity of population, and good genes among populations have been exchanged, which is conducive to evolving global optimal solution. When the group number N continues to increase, because the population number is certain, the more the group number is, the less the individual of each sub population is, which reduces the search accuracy and stability. So it is not conducive to the evolution.

Fig. 7 shows that when the evolution generation T is 400, the average fitness value of the final individual is the highest, which satisfies the design requirements. When the evolution generation T decreases, because the sub populations are not fully evolve, it is not conducive for the hybrid strategy and the final result. When the evolution generation T increases, because the sub populations have already evolved to the optimal individual, resources and time are waste to continue to evolve. The genetic operatior can also affect the optimal individual gene, and it is not be conducive to the whole evolution and the final result.

Conclusions

In view of the evolvable hardware, the current genetic algorithm has a long evolutionary time, slow convergence speed, and a large amount of computation. In order to improve algorithm and solve the current problems. This paper proposed an improved genetic algorithm--stage hybrid genetic algorithm, whose inspiration was from the biological population hybridization, and offspring individuals inherited good traits form different parents.

Using evolvable hardware platform, improved algorithm adopted the stages strategy and simulated the natural population hybridization based on genetic principle, getting offspring circuits more adaptive and close to the global optimum. In the performance of average circuit fitness, the comparative evolution results show that improved algorithm has better individual circuits than standard algorithm, and offspring circuits have a higher adaptability. The stages strategy shortens evolutionary period and improves evolutionary efficiency. The hybrid strategy expands the search scope, increases the population diversity, and prevents effectively the occurrence of local optimal solutions. The selection strategy improves the population convergent rate, which makes the search more accurate; In the Sorting comparison of two algorithms, the results show that the improved algorithm has less average comparison number and lower average time consuming, which reduces evolutionary computation and carries out parallel computing to save system resources; Finally, the grouping number N and evolutionary generation T of improved algorithm are analyzed, finding the optimal parameter values. Algorithm accuracy is improved, which is beneficial to the whole evolution.

References

[1] Xin Yao, T. Higuchi. Promises and Challenges of Evolvable Hardware. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews. 1999, 29(1): 87-97.

[2] Jixiang Zhu, Yuanxiang Li, Xuewen Xia. Online adaptive systems based on evolvable hardware. Computer Science, 2009, 36(7): 267-269.

[3] Youren Wang, Yuanyuan Huang, Ran Feng. Logic circuit evolutionary design based on matric encoding. Chinese Journal of Electronics, 2011, 39(11): 2576-2582. In Chinese.

[4] J.D. Lohn, S.P. Colombano. A circuit representation technique for automated circuit design[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(3): 205-219．

[5] K.V. Vesselin, F.M. Julian. The advantages of landscape neutrality in digital circuit evolution. Lecture Notes in Computer Science, 2000, pp. 252–263.

[6] Zeyu Sun, Tao Yang, Yunxing Shu. Wireless sensor networks optimization covering algorithms based on genetic algorithms. Computer Modelling and New Technologies, 2014, 18(4): 50–56.

[7] A.P. Shanthi, Ranjani Parthasarathi. Practical and scalable evolution of digital circuits. Applied Soft Computing, 2009, 9(2): 618-624.

[8] Huicong Wu. Fault tolerant circuit design using evolutionary algorithms. Journal of Computers, 2014, 9(1): 95-100.

[9] Yan Li, Vladimir Stojanovic. Yield-driven Iterative Robust Circuit Optimization Algorithm. Design Automation Conference, 46th ACM/IEEE, 2009, pp. 599-604.

[10] Huicong Wu, Jinze Wang, Chuncao Liu, et al. Research of circuit evolution design based on adaptive HereBoy algorithm. Journal of Heibei University of Science and Technology, 2015, 36(3): 293-299. In Chinese.