Application Research of Evolutionary Algorithm in Synthesis of Reversible Logic Circuits

Jiaxin Han, Xin Zhang, Xiaoxiao Wang

School of Computer Science, Xi'an Shiyou University, Xi'an 710065, China
zx13679175975@sina.com

Abstract. In order to avoid the algorithm falling into premature convergence, a big mutation adaptive evolutionary algorithm for reversible logic circuit synthesis with tabu list is proposed. Big Mutation Variable Length Evolutionary Algorithm Reversible Logic Circuit (BM_VLEA_RLC) is based on Variable-Length Evolutionary Algorithm Reversible Logic Circuit (VLEA_RLC) algorithm that designs the corresponding fitness function, adaptive evolution probability, etc., and tests using benchmark functions. The experimental results show that the proposed algorithm can obtain several feasible solutions, and some solutions are better than the existing algorithms, which verifies the feasibility of the algorithm.

1. Introduction

Laudauer[1] points out that the energy consumption of IC chip mainly originates from the irreversible operation of the signal calculation in chip. Therefore, the key to reduce chip energy consumption is turning the irreversible operation into reversible operation. Reversible circuit synthesis is to select a series of reversible gates in the existing standard reversible logic gate library to form a circuit, so that it logically satisfies the specification of reversible functions, both input and output mapping relationship while ensuring that the circuit costs lowly[2].

At present, the evolutionary algorithm is widely used in the field of quantum and reversible circuit synthesis because of its simple operation process, no limitation of derivation, and global search ability. In this aspect, Wang et al, proposed the VLEA_RLC[3] algorithm, which minimizes the quantum cost of the circuit without introducing the garbage bits. On the one hand, the conflict of constraint violation and object value can be well handled by the improved stochastic ranking method. On the other hand, the qualification factor information extracted from the reversible function Positive Polar Reed-Muller (PPRM) expression is applied to the evolutionary algorithm. Compared with the traditional evolutionary algorithm, VLEA_RLC algorithm can effectively control chromosome expansion and improve the ratio of feasible solutions. However, the fixed crossover and mutation probability used in VLEA_RLC algorithm can easily fall into premature convergence. In this paper, the VLEA_RLC algorithm is improved, which is called Big Mutation Variable Length Evolutionary Algorithm Reversible Logic Circuit (BM_VLEA_RLC) algorithm. The improvements are mainly reflected in the following two points:

• The crossover and mutation probability are changed according to individual fitness.
• Use big mutation operation and tabu list.

The structure of this paper is as follows. Section 2 introduces the proposed algorithm. Section 3 describes the experimental research of the algorithm. Section 4 summarizes the paper.
2. The Proposed Algorithm

Firstly, some improvements are pointed out in this paper, then, some key techniques used in the proposed algorithm are introduced, and finally the details of the proposed algorithm are given.

The fixed crossover and mutation probability used in VLEA_RLC algorithm can easily fall into premature convergence. In this paper, two improvements are proposed for the shortcomings of VLEA_RLC.

The first is to change the crossover and mutation probability base on the fitness of the individual. Improve the convergence precision and speed up the convergence speed. The second is using big mutation operation and tabu list when a generation of all individuals are falling into premature convergence. Using big mutation operation can randomly produce many new individuals, in order to the whole population out of premature convergence. Using the tabu list is to avoid the algorithm falling into the same local optimal solution.

The proposed algorithm is mainly divided into six modules: coding scheme, fitness function, crossover operator, mutation operator, big mutation operation and tabu list. These modules are described in detail in the following sections.

2.1. Coding scheme

The coding scheme adopts the coding scheme proposed by Wang[3]. The scheme uses the structure module sequence to represent a series of gates that make up the circuit. The structure module sequence includes a series of tuple consist of control bit and object bit.

2.2. Fitness function

The primary task of quantum reversible circuit synthesis is to select a series of reversible gates in the existing standard reversible logic gate library to form a circuit, so that it logically satisfies the specification of reversible functions. Using the fitness evaluation to detect the difference between current individual and specification function. The deviation value $error$ and the fitness function $fit$ are defined as follows:

$$error = |pprm_k - pprm_i|$$

$$fit = \frac{1}{1 + error}$$

Where $pprm_k$ is an identity function PPRM expression, $pprm_i$ is an individual PPRM expression. The error of the circuit is defined as the number of different items between the PPRM expression of the simplification function and the PPRM expression of the identity function. The smaller the error, the larger the fit.

2.3. Crossover operator

Cross operation is the key way to produce a new population, the parent generates new individuals by cross-swap some genes. In this paper, Synapsing Variable Length Crossover (SVLC)[4] is used to possibly inherit the parent public sub-module and improved quality of the solution as far as. The crossover probability can adapt to the fitness of the individual and increase the diversity of the population. The crossover probability is expressed as follows:

$$p_c = k_1 * (1 - fit_i)$$

Where $k_3$ is constant, $fit_i$ is the fitness of the current individual.

2.4. Mutation operator

The mutation operation is to replace some gene values in the individual coding with other gene values to form a new individual. It can enhance the local search ability of the algorithm and maintain population diversity[5]. In order to achieve rapid circuit synthesis, any two genes in a chromosome and exchange their location, the control and object bits of the mutant gene and the deletion of certain genes are applied. The mutation probability changes with the individual fitness. Its design principle is similar to the crossover operator and the mutation probability is expressed as follows:
\[ p_m = k_2 \ast (1 - \text{fit}_i) \]  \hspace{1cm} (4)

Where \( k_2 \) is a constant, \( \text{fit}_i \) is the fitness of the current individual.

2.5. Big mutation operation

When a generation of all individuals is concentrated, the diversity is lost. Many new individuals are produced by big mutation operation randomly and independently. It can improve the whole population out of premature convergence and the specific operation process is as follows:

\[ \alpha f_{\text{max}} < f_{\text{avg}} \]  \hspace{1cm} (5)

Where \( f_{\text{max}} \) represents the maximum fitness of a generation, \( f_{\text{avg}} \) represents average fitness, \( \alpha \) is the denser factor and this is used to indicate the population density, the larger \( \alpha \), the more concentrated the individuals and the lower the population diversity. The value of \( \alpha \) is usually given at the beginning of the algorithm. When equation (5) is satisfied, the mutation probability of all individuals in the population is increased to several times the previous mutation probability and then the whole population makes the operation of mutation.

2.6. Tabu list

With the principle of tabu search[6], set up tabu list to store all the local optimal solutions in the recent history of the search process, the algorithm falling into the same local optimal solution repeatedly would be avoided. During the algorithm operation, when the population makes big mutation, the solution corresponding with the highest fitness is added to the tabu list. When the value of an individual in a generation is the same as the value in the tabu list, the corresponding individual fitness is reduced that prevents the algorithm from falling back into the historical local optimal solution and prevents the cyclic search of the algorithm.

2.7. The framework of BM_VLEA_RLC

The proposed method is based on VLEA_RLC algorithm, which changes the crossover and mutation probability through individual fitness, using the big mutation operation and the tabu list to prevent the algorithm from premature convergence. The BM_VLEA_RLC algorithm is presented as shown in Algorithm.

Algorithm: require \((pprm,n,k_1,k_2,T,p_f,\alpha,p,\text{iter},g_{\text{max}},p_{\text{max}})\)

Inputs: \( pprm \) : PPRM expression; \( n \) : population size; \( k_1 \) : cross probability; \( k_2 \) : mutation probability; \( T \) : population update interval; \( \text{iter} \) : number of iterations; \( \alpha \) : dense factor; MSR algorithm parameters: \( p_f \) : probability factor; \( p \) : pressure parameter; \( g_{\text{max}} \) : max generations; \( p_{\text{max}} \) : big mutation probability;

Outputs: \( R \) optimal reversible circuit set

1. let input PPRM expression, get factor, the maximum number of control bits and other information, create circuit door library, estimate the length of the individual
2. let initial population \( P(t), t = 0, \text{size} = n, R = \{\} \)
3. let calculate an individual error value and cost value
4. let individual sorting using the MSR[3] algorithm
5. let \( \text{min} \_\text{cv} = 10000, \text{min} \_\text{cost} = 10000, u = 0, S = \{\} , p_{\text{max}} \), use \( \text{min} \_\text{cv} \) to record the minimum error, \( \text{min} \_\text{cost} \) to record the minimum cost, \( k \) to record the individual, \( u \) to record the number of times that the historical optimal solution has not been updated, and \( S \) to represent tabu list.
6. for each \( i \) in \( \text{iter} \):
   7.   for each \( j \) in \( g_{\text{max}} \):
      8.     if \( \text{cv}_k < \text{min} \_\text{cv} \) or \( \text{cost}_k < \text{min} \_\text{cost} \) then
      9.        let \( \text{min} \_\text{cv} = \text{cv}_k, \text{min} \_\text{cost} = \text{cost}_k \)
     10.    if \( u > T \) then
       11.        let perform population update and \( u = 0 \)
12.  ```
13.  else then
14.  if \( \alpha f_{max} < f_{avg} \) then
15.  add(max_\(f_\), \(f_{max}\)) to \(S\) and use \(p_{max}\) to perform big mutation.
16.  ```
17.  ```
18.  ```
19.  ```
20.  ```
21.  ```
22.  ```

3. Analysis of experimental results

To evaluate the effectiveness of the algorithm, it is applied on some benchmark functions and the results are compared with the known optimal results. These optimal results are clearly specified in the corresponding literature.

Indexes of algorithm evaluation are the circuit gate number \(g\) and the quantum cost \(c\). The circuit gate number refers to the number of reversible gates forming the circuit and the quantum cost refers to the sum of costs forming the circuit.

The algorithm parameters are set as follows: \(\alpha\) represents dense factor is 0.7, \(p_{max}\) represents big mutation probability is 0.6, \(k_1\) represents cross factor is 0.8, \(k_2\) represents mutation factor is 0.15, \(n\) represents population size is 500, \(T\) represents population update interval is 200, \(p_f\) represents probability parameter is 0.3, \(p\) represents pressure parameter is \(2^{\text{maxConNum}} - 3\), where \(\text{maxConNum}\) is the maximum number of control digits, \(\text{iter}\) represents the number of iterations is 30, and \(g_{max}\) represents the maximum evolutionary generations is 2000.

Table.1 Experimental results of different algorithm comparisons.

| Function | Known GC Minimization Circuit | Known QC Minimization Circuit | The Algorithm |
|----------|-------------------------------|-------------------------------|---------------|
|          | \(g\) | \(c\) | literature | \(g\) | \(c\) | literature | \(g\) | \(c\) | literature |
| 3_17     | 6    | 14    | [3]         | 6    | 14    | [3]         | 6    | 14    | [3]         |
| 4_49     | 12   | 32    | [3]         | 12   | 32    | [3]         | 12   | 32    | [3]         |
| hwbc4    | 11   | 23    | [3]         | 11   | 23    | [3]         | 11   | 23    | [3]         |
| xor5     | 4    | 4     | [7]         | 4    | 4     | [7]         | 4    | 4     | [7]         |
| rd32     | 5    | 15    | [8]         | 4    | 8     | [3]         | 4    | 8     | [3]         |
| ham3     | 5    | 9     | [10]        | 5    | 7     | [3]         | 5    | 7     | [3]         |
| ham7     | 21   | 65    | [10]        | 22   | 49    | [3]         | 21   | 65    | [3]         |
| rd53     | 13   | 116   | [10]        | 12   | 36    | [3]         | 12   | 36    | [3]         |
| majority5| 16   | 104   | [11]        | 16   | 68    | [3]         | 16   | 68    | [3]         |
| mod5adder| 15   | 83    | [9]         | 17   | 69    | [3]         | 16   | 72    |               |
| mod15adder| 10  | 71    | [11]        | 13   | 57    | [3]         | 10   | 57    |               |
| mod32adder| 15  | 154   | [11]        | 18   | 134   | [3]         | 16   | 136   |               |

It can be seen from Table.1, the algorithm can obtain the feasible solution of 12 standard test functions that run independently for 30 times. In some test functions, better quantum cost \(c\) and less gate circuit number \(g\) can be obtained, the feasible solution rate on 12 functions is 1, and the same
value is obtained on 9 functions as the known optimal result. A smaller object value is obtained on 3 functions. Experimental results show that the proposed algorithm can obtain lower quantum cost on small and medium scale problems.

4.  Conclusion
Based on the VLEA_RLC algorithm, this paper proposes a new BM_VLEA_RLC algorithm. Compared with the results of the existing algorithms, the proposed algorithm achieves better results on some test problems. It has a more uniform population distribution and better local search ability. The emphasis of further research will be how to apply the algorithm to the medium and large scale circuit problems and improve the local search ability.

Acknowledgments
This work was financially supported by the Natural Science Basic Research Plan in Shaanxi Province of China (2018JM6093), Scientific Research Plan of Shaanxi Committee of Education (17JK0595), and Shaanxi Provincial International Co-operation and Exchanges in Science and Technology Plan Project (2016kw-047).

References
[1] Landauer R. Irreversibility and heat generation in the computing process [J]. IBM journal of Research and Development, 1961, 5(3): 183-191.
[2] Saeedi M, Markov I L. Synthesis and optimization of reversible circuits—a survey [J]. ACM Computing Surveys (CSUR), 2013, 45(2): 21
[3] Wang Xiaoxiao, Research on evolutionary reversible logic circuit synthesis method [D]. Xi’an University of Electronic Science and Technology, 2016 (in Chinese).
[4] Mallipeddi R, Suganthan P N. Ensemble of constraint handling techniques [J]. IEEE Transactions on Evolutionary Computation, 2010, 14(4): 561-579.
[5] Luo Xiao. Research on reversible logic synthesis method based on multi-chromosome gene expression programming [D]. Donghua University, 2017 (in Chinese).
[6] Zhang Zhen, Wei Peng, Li Yufeng, Lan Julong, Xu Ping, ChenN Bo. Feature selection algorithm for improved particle swarm combined tabu search [J]. Journal on Communications, 2018(12):60-68 (in Chinese).
[7] Wang Xu. Improved Differential Evolution Algorithms and Their Applications in Reversible Logic Synthesis [D]. Donghua University, 2013 (in Chinese)
[8] HU Jiang, ZHANG Qiaowen, WANG Yang. Quantum reversible circuits synthesis based On improved genetic algorithm [J]. Chinese Journal of Quantum Electronics, 2017, 34:196-202 (in Chinese).
[9] Golubitsky O, Maslov D. A study of optimal 4-bit reversible toffoli circuits and their synthesis [J]. IEEE Transactions on Computers, 2012, 61(9): 1341-1353
[10] Zhang Mingming. Research on Multi-objective Evolutionary Algorithms for Quantum Reversible Logic Automatic Synthesis [D]. Donghua University, 2010 (in Chinese).
[11] Gupta P, Agrawal A, Jha N K. An algorithm for synthesis of reversible logic circuits [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 25(11): 2317-2330