A Circuit Generating Mechanism with Evolutionary Programming for Improving the Diversity of Circuit Topology in Population-Based Analog Circuit Design

Mei Xue and Jingsong He*

Department of Electronic Science and Technology,
University of Science and Technology of China, Hefei, China, 230027
jxue@mail.ustc.edu.cn, hjss@ustc.edu.cn

Abstract. This paper presents an analog circuit generating mechanism based on connecting point guidance existing in circuit netlist. With the proposed mechanism, the initial circuit topology can be a random netlist, and the evolutionary operation can be executed directly on connecting point. Also, the knowledge of graph theory is introduced for evaluating the degree of diversity of circuit structures. Experimental results show that the proposed mechanism is beneficial to improve the diversity of topology in population. In the case of no robustness evolution mechanism, the diversity of topology in population can improve the fault tolerance of population.

Keywords: Analog circuit, circuit topology, evolutionary programming.

1 Introduction

Analog circuit module plays an important role in many electronic systems. Unlike digital circuit design, analog circuit design doesn't have nature CAD tools. The task of analog circuit design is very complicate and the final result largely depends on the designers' knowledge and experience. In the past, researchers pay many attentions to analog circuit design automation with the application of artificial intelligence [1-9]. Koza et al. [1] propose a tree-coded scheme with genetic programming, by which the circuit topological structure is determined by the tree storage structure and connection-modifying functions. Lohn at el. [2] propose a linear-coded mechanism combining with genetic algorithm, which has been successfully used in evolutionary design of analogy filters and amplifiers. Grimbleby [3] propose a kind of netlist-based representation method which can directly generate circuit with a few restriction on the topology structures of circuits. Zebulum at el. [4] apply the netlist-based representation to the synthesis of circuits with three-terminal component. [5-6] limit the number of components, where the number of components is different from the length of chromosomes.

* Corresponding author.

Y. Tan, Y. Shi, and H. Mo (Eds.): ICSI 2013, Part I, LNCS 7928, pp. 540–547, 2013.

In all the above methods, encoding method presented in [1] is complicated, the connection type of component is set in advance in [2], and the circuit topological structure is determined by connection type. Since a certain amount of connections cannot include all kinds of topologies, the encoding method has limitation on circuit topological structure. The netlist-based encoding mechanism directly generates the topology of circuits by connecting points. Topologies generated by this method are more flexible, thereby the method is helpful to get rich topologies. However, the length of chromosomes in [3-6] is fixed. This situation is not conducive to the diversity of population.

This paper presents a connecting point guidance circuit generating mechanism. It constructs the topology of a circuit by all its connecting points. The number of components in a circuit is determined both by the number of connecting points and the number of components connected to every connecting point. It directly operates the connecting points by mutation operators during evolution. These mutation operators offer a more flexible approach to obtain rich topologies. Experimental results show that the connecting point guidance circuit generating mechanism is beneficial to improve the diversity of topologies in population. In the case of no robustness evolution mechanism, the diversity of circuit topologies in population can improve the fault-tolerance of population.

2 Circuit Generating

This new circuit generating mechanism is based on connecting point of circuits, where the number of components in a circuit is determined both by the number of connecting points and the number of components connected to every connecting point. In this way, the number of components of the candidate circuits will be changeable and flexible, on condition that these two parameters have not been fixed at the stage of initialization. Genetic operators are directly associated with encoding method, we adopt five kinds of mutation operators which specially designed for connecting point guidance circuits generating mechanism. These five mutation operators are complete to the evolutionary process, and can produce any circuit structure.

2.1 Netlist Generating

Netlist-based encoding method of circuit was firstly proposed by Grimbleby in [3]. The netlist-based encoding method we used here is different from previous ones. Our method is to control the number of components in a circuit by the number of connecting points and the number of components connected to every connecting point. It is clear, if these two parameters are fixed, the number of components in circuit are same in initialized population. As a solution, we can get diverse individuals by setting the range of the two parameters. We set a parameter *MC* to control the number of connecting point, and another parameter N_i to control the number of components that the *i-th* connecting point connected. When these two parameters are variable, the number of component in every circuit is unfixed. On the basic of embryonic circuit (see left in Fig. 1), 0 represents ground, 1 represents input and 255 represents output, connecting points in a circuit are [0,1,2,..., MC,255]. The length of the circuit is L ,

$$L = N * \left\lceil \frac{MC+2}{2} \right\rceil \quad . \tag{1}$$

Assume that N of a circuit is from 2 to 5, then

$$2 * \left\lceil \frac{MC+2}{2} \right\rceil \le L \le 5 * \left\lceil \frac{MC+2}{2} \right\rceil \quad . \tag{2}$$

If the range of MC is from 4 to 10, the length of the circuit is from 6 to 30.

The right figure in Figure 1 shows the map of a random netlist. We take a connecting point connect at least two not parallel components in whole circuit as guiding principle to avoid invalid circuits as much as possible. We use real number encoding technique to initialize population, e.g., 1 represents resistance, 2 represents capacitance, and 3 represents inductance. Thus, the value of every component can randomly be selected by its type and the corresponding range of value.

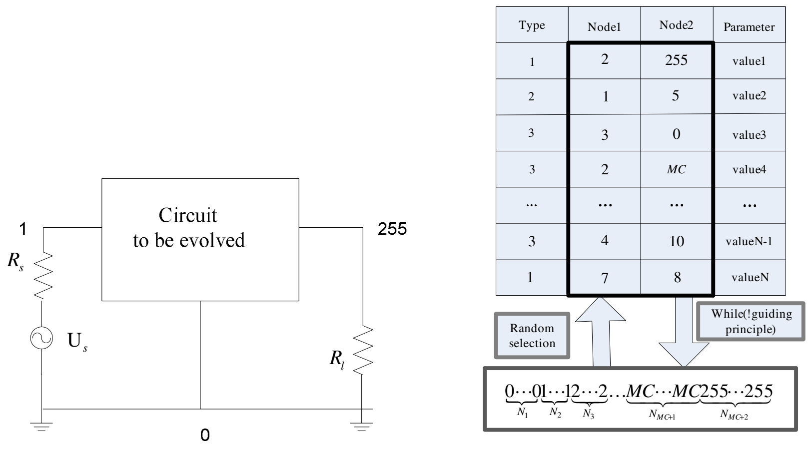

Fig. 1. Left: the embryonic circuit of analog filter ; right: the process of netlist generating

2.2 Mutation Operations

We adopt five mutation operators according to the actual requirements. These five mutation operators are enough to generating any kinds of topologies.

1. parameter change: select a component randomly, replace its value with a new one which randomly select from the range of parameter.

2. type change: select a component randomly, the type is changed to a different one and the parameter is changed simultaneously.
3. point change: randomly select two different connecting point in circuit netlist, then swap them.
4. component adding by point: select two different connecting points randomly in circuit netlist, connect the two connecting points by one component, its type and value are randomly selected.
5. component deletion: delete a component which is randomly selected from circuit netlist by merging the two connecting points into one.

In these five mutation operators, the last three are specially designed for connecting point, they can flexibly operate circuit topology, and can produce circuits with any topology. Mutation operation is likely to produce invalid individuals in evolutionary process , we take a connecting point connect at least two non-parallel components in an whole circuit as guiding principle to avoid invalid circuits as much as possible.

2.3 The Measurement of Circuit Topology

It is said that small world patterns exist in electronic circuits in [10]. Taking circuits as graphs, graphs with a small world structure are highly clustered but path length will be small [10]. Here, we choose clustering coefficient to describe circuit topology. We simplify circuits by series and parallel until their topologies contains only nodes, the node is the connecting point which connect three or more than three components.

Assume that node *i* has *k* edges to other nodes, the maximum number of edges between the k_i nodes is $C_{k_i}^2$, and the actual number of edges is E_i . The clustering coefficient of a circuit is the average of all nodes, i.e.,

$$C_3 = \frac{1}{N}\sum_{i=1}^{N}\frac{E_i}{C_{k_i}^2} \quad . \tag{3}$$

From the perspective of geometry, the clustering coefficient of node *i* is the number of triangles connected to node *i* relative to the number of ternary group with the center of node *i*.

It can be found that the clustering coefficient above can only reflect the topology of triangle, not include the quadrilateral topology. For this reason, we have

$$C_4 = \frac{1}{N}\sum_{i=1}^{N}\frac{E2_i}{C_{k_i}^3} \quad . \tag{4}$$

where, $E2_i$ denotes the number of quadrilaterals connected to node *i.* The measurement coefficient of topologies can be $C = \alpha C_3 + \beta C_4$, and $\alpha + \beta = 1$. In this paper, we have $\alpha = \beta = 0.5$.

3 Evolutionary Method

The search engine we adopt in this paper is evolutionary programming [11], it is a kind of evolutionary algorithm based on population. The number of initialized individuals is *NP*, every individual produce a child after mutation operation, every time one mutation operator is randomly selected to operate a individual. Each iteration we select ten percent of optimal individuals from parents and children, others are selected by stochastic tournament mechanism. The process of analog circuit design is described as follow:

1. Randomly generate the initial population of *NP* individuals with the connecting point guidance circuit generating mechanism, each circuit is stored in a two-dimensional array just like the right figure in Figure 1, X_n denote the *n-th* individual, $\forall n \in \{1,\cdots,NP\}$.
2. Evaluate the fitness of each individual in the population based on the fitness function in [1].
3. Each individual X_n, n = 1,…,*NP*, produce a offspring X_n by a kind of mutation operators which are randomly selected from five mutation operators with equal probability.
4. Calculate the fitness of each offspring $X_n^{'}$ as step 2, $\forall n \in \{1,\cdots,NP\}$.
5. Select $0.1*NP$ optimal individuals out of $X_n \cup X_n^{'}$, $\forall n \in \{1,\cdots,NP\}$, the other $0.9*NP$ individuals are selected by q stochastic tournament mechanism in remaining population. For each individual a_i, $\forall i \in \{1,\cdots,1.9*NP\}$ in remaining population, q opponents are chosen randomly with equal probability, $score_i$ is the number of individuals whose fitness are bigger than a_i. After the process of comparison, sort the individuals by the rising trend of their scores. Then select $0.9*NP$ individuals with high scores to the parents of the next generation.
6. Stop if the halting criterion is satisfied; otherwise, go to step 3.

4 The Experiments and Results

We take analog filter as experimental observation object. The goal of experiments is to automatically design analog lowpass filters using evolutionary programming. The transition zone is from 1000 *Hz* to 2000*Hz*, the ripple of passband is 30*mV* or less and the ripple of stopband is 1*mV* or less.

There are many kinds of faults, they can be roughly divided into two classes, parameter perturbations and topology failures. We use topology failure model to test every component in circuits with single point of short and disconnection damages[8]. A crucial-component in a circuit is the component whose failure will result in the losing of its original function.

4.1 Filter Design

Population size is set to 200 individuals, and each run proceeds for 400 generations. The results shown in Figure 2 are chosen from ten runs. Top left shows the best fitness through generation, fitness of four runs decrease gradually indicate that populations are convergent. Top right shows a schematic of lowpass filter, it is satisfied the specification and randomly selected from four runs. From the schematic we can see that the circuit evolved by the connecting point guidance circuit generating mechanism contains novel topology, rather than the traditional filters contain only T-shaped and π-shaped topology. Bottom left shows the change of circuits' length in population in four runs, we can see that the number of component in population is changeable and flexible. Bottom right shows the change of circuit structure in population in four runs, we can see that there are a variety of topologies in population. These experiments illustrate that the connecting point guidance circuits generating mechanism proposed by this paper can be a potential assistant method to design analog circuits. It is beneficial to improve the diversity of topology in population and produce novel circuit topology.

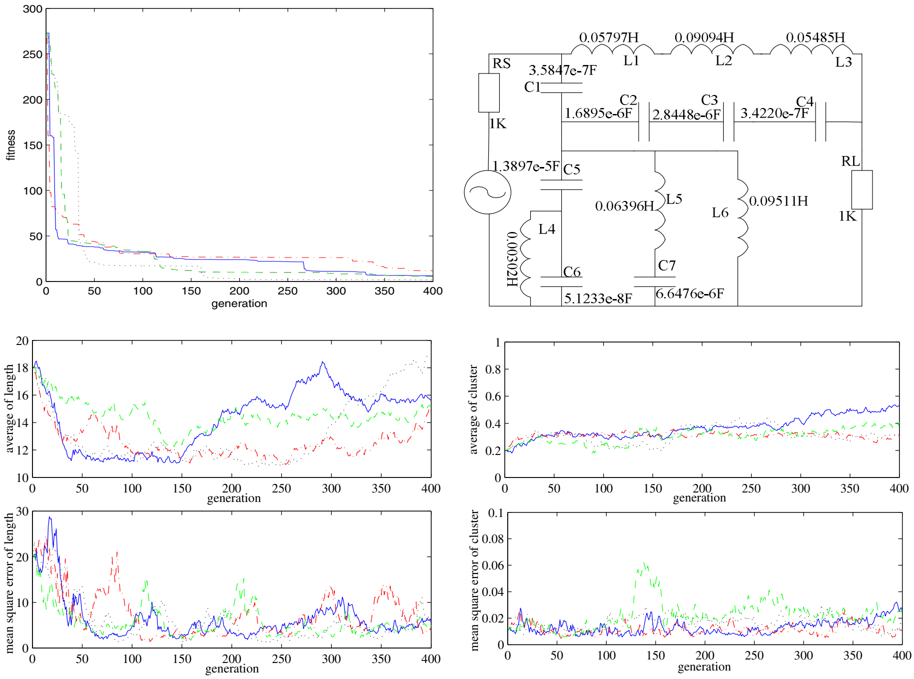

Fig. 2. The results of four runs which chosen from ten runs in lowpass filter design. Top left: Fitness of best circuit through generation in four runs; Top right: a circuit randomly selected which satisfy the specification. Bottom left: the average and the mean square error of circuits' length in four populations; Bottom right: the average and the mean square error of measurement coefficient in four populations; The same color represents the same population

4.2 Fault Tolerance Test

The statistics of fault tolerant performance of individuals in population can better indicate the fault tolerance of population, so we can choose some individuals which meet the design requirements from populations, then test their fault tolerance. Suppose $f_{tolerance}$ is the fault tolerance, it is the sum of single point fault model test in Eq. (5). Assume that the number of component in a circuit is n, and T is a threshold, if the fitness of single point fault model test is bigger than T, the weight of fitness is 0.1, else the weight is 0.01.

$$f_{tolerance} = \frac{w}{n} \sum_{i=1}^{n} fitness\ (i) \ . \tag{5}$$

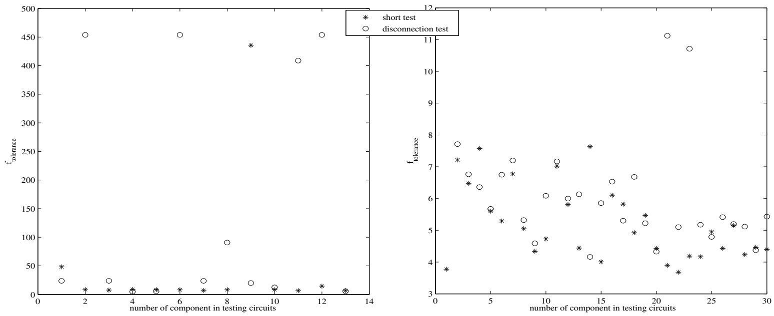

Fig. 3. The results of fault tolerance test. Left: the fault-tolerance performance of top right with single point of shorts and disconnection damages; Right: the fault tolerance of 30 individuals randomly selected in ten runs with single point of shorts and disconnection damages, and they all match the specification.

In Figure 3, left reflects the fault tolerance of top right in Figure 2, we can see that there is one crucial-component in circuit when we have single point short damage, and four crucial-components with disconnection damage. The $f_{tolerance}$ of single point short damage test is 3.7770, and the $f_{tolerance}$ of single point disconnection damage test is 11.6282, the circuit's fault tolerance to disconnection damage is worse than short damage.

We choose 30 different individuals which meet the specification from ten runs, then test them with single point of short and disconnection damages respectively. It can be seen from right in Figure 3 that most individuals have some degree of fault tolerance. The circuits are randomly selected, so we have reasons to believe that rich topologies can bring a certain degree of fault tolerance to population.

5 Conclusion

In this paper, we propose a connecting point guidance circuit generating mechanism, this mechanism coordinates with evolutionary programming can evolve analog circuits. This circuit generating mechanism almost has no limit on circuit topology, circuits evolved by this method have various number of components. The experimental results shows that the connecting point guidance circuit generating mechanism benefits to improve the diversity of topologies in population. In the case of no robustness evolution mechanism, rich topologies can bring a certain degree of fault tolerance to population. In the following work, we would use this circuit generating mechanism in the evolution of circuits which contain three-terminal components, even circuits with multiport modularized circuit, and make connecting point guidance circuit generating mechanism become a universal assist to netlist-based representation in analog design automation.

Acknowledgments. This work is supported by National Nature Science Foundation of China under Grant 60975051 and 61273315.

References

1. Koza, J.R., Benett III, F.H., Andre, D., et al.: Automated synthesis of analog electrical circuits by mean of genetic programming. IEEE Transactions on Evolutionary Computation 1(2), 109–128 (1997)
2. Lohn, J.D., Colombano, S.P.: A Circuit Representation Technique for Automated Circuit Design. IEEE Transactions on Evolutionary Computation 3(3), 205–219 (1999)
3. Grimbleby, J.B.: Automatic Analogue Network Synthesis Using Genetic Algorithms. In: The First Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, Sheffield, UK, pp. 53–58 (1995)
4. Zebulum, R.S., Pacheco, M.A., Vellasco, M.: Artificial Evolution of Active Filters: A Case Study. In: The First NASAIDOD Workshop on Evolvable Hardwar (1999)
5. Ando, S., Iba, H.: Analog circuit design with a variable length chromosome. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 2, pp. 994–1001 (2000)
6. Goh, C., Li, Y.: GA automated design and synthesis of analog circuits with practical constraints. In: Proc. of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 170–177 (2001)
7. Sapargaliyev, Y., Kalganova, T.: Constrained and unconstrained evolution of "LCR" low-pass filters with oscillating length representation. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1529–1536 (2006)
8. Kim, K.J., Wong, A., Lipson, H.: Automated synthesis of resilient and tamper-evident analog circuits without a single point of failure. Genet Program Evolvable Mach 11, 34–59 (2010)
9. He, J., Zou, K., Liu, M.: Section-representation scheme for evolutionary analog filter synthesis and fault tolerance design. In: Advanced Computational Intelligence (IWACI), vol. 3, pp. 265–270 (2010)
10. Ferrer, I., Cancho, R., Janssen, C., Sole, R.V.: Topology of technology graphs: Small world patterns in electronic circuits. Physical Review 64, 461 (2001)
11. Yao, X., Liu, Y.: Fast evolutionary programming. In: The Fifth Annual Conference on Evolutionary Programming (1996)