

Chaotic binary bat algorithm for analog test point selection

Dongsheng Zhao¹ · Yuzhu He¹

Received: 10 February 2015 / Revised: 25 March 2015 / Accepted: 18 April 2015 / Published online: 24 April 2015
© Springer Science+Business Media New York 2015

Abstract A set of novel methods are proposed to optimize analog test point selection in this paper. We have introduced chaos and elitist strategy into the binary bat algorithm so as to increase its global search mobility and effectiveness for robust global optimization. In the present study, nine well known chaotic maps are introduced and used to construct the chaotic binary bat algorithm (CBBA) respectively. As a result, we have developed nine different CBBA and for the first time applied them to deal with the analog test point selection problem. The attractiveness of our proposed CBBA mainly lies in two aspects: the utilization of chaotic maps to tune the BBA parameter and the application of elitist strategy to store all the possible global best solutions. These improvements obviously enhance the performance of BBA and make our new algorithms (CBBA) get all the best solutions (usually more than one) easily and effectively, which will give us more possible choices in practice. Analog circuits' examples and a group of statistical experiments are given to demonstrate the feasibility and effectiveness of the proposed algorithms. The other reported algorithms are also used to do the comparison. The results indicate that the proposed algorithms have excellent performance in finding the optimum test point sets. Therefore, they are good solutions and applicable to actual circuits and engineering practice.

Keywords Analog fault diagnosis · Test point selection · Chaotic maps · Bat algorithm (BA) · Binary bat algorithm (BBA) · Chaotic binary bat algorithm (CBBA)

✉ Dongsheng Zhao
mailzds@126.com

¹ School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China

1 Introduction

The test point selection is becoming more and more important in analog circuit fault diagnosis nowadays. The analog circuit diagnosis methods are usually classified into two main categories [1–3]: the simulation before test (SBT) and the simulation after test (SAT) approach. Fault dictionary and probabilistic techniques fall into SBT, whereas optimization, fault verification, and parameter identification techniques fall into SAT [2]. Since each circuit under test (CUT) consists of more than two test points, especially for the medium and large scale circuits, it's impractical and too expensive to test the responses of all the test points to diagnose the faulty circuit. At the same time, not every test point is measurable and some measurements are redundant. Therefore, the optimum selection of test points is especially significant.

The fault dictionary technique has been proven to be a very practical method in analog catastrophic fault diagnosis [3]. A fault dictionary is a set of measurements of the CUT simulated under potentially faulty conditions (including fault-free case) and organized before the test. The measurements could be at different test points, test frequencies, and sampling times [4, 5]. There are three important phases in the fault dictionary approach [5, 6]. First of all, a network is simulated for each of the anticipated faults (including fault-free case) excited by the chosen stimuli (dc or ac), and the signatures of the responses are stored and organized in the dictionary for use. The second phase is the selection of test points. An optimum selection of test points is the main work of this stage. By doing this, we can achieve the desired degree of fault diagnosis with less test points and save the fault test and diagnosis time greatly. The last phase is fault isolation. At the time of testing, the CUT is excited by the same stimuli that are used in

constructing the dictionary, and measurements are made at the preselected test points. They are compared with the responses stored in the fault dictionary to identify the fault according to the preset criteria. This paper mainly focuses on the second phase.

The aim of the analog test point selection is to find the least number of test points that can isolate all the faults. Until now, lots of researchers have been studied in this area. Varghese et al. [7] proposed a heuristic method based on given performance indexes to find the sets of test points. Hochwald and Bastian [8] proposed the concept of ambiguity sets and developed logical rules to select the test points. Lin and Elcherif [3] first proposed the integer-coded dictionary to distinguish the ambiguity sets for analog test point selection. Spaandonk and Kevenaar [9] proposed to select the test-point set by combining the decomposition method of the system's sensitivity matrix and an iterative algorithm. Prasad and Babu [6] proposed four algorithms based on three strategies of inclusive approach and three strategies of exclusive approach. Pinjala and Bruce [10] proposed a method to find the test point set by computing the information content of all the candidate test points. An entropy-based approach, using entropy index (EI) as a criterion to choose test points, was proposed by Starzyk et al. [4]. Golonek and Rutkowski [11] used a genetic algorithm (GA) based method to determine the optimal set of test points. Yang and Tian [5] used the graph node search method to find the near minimum test-point set. Jiang et al. [12] proposed a multidimensional fitness function discrete particle swarm optimization (MDFDPSO) algorithm to optimize analog test point selection. Luo and Wang [13] used the extended fault dictionary and the overlapped area values together to select the optimum test points.

In 2010, Yang [14] proposed the bat algorithm (BA), which is a novel heuristic optimization algorithm based on the echolocation behavior of bats. The capability of echolocation of bats is so special and powerful that they can find their prey and discriminate different types of insects even in complete darkness. Preliminary studies suggest that the BA can have superior performance over particle swarm optimization (PSO) and genetic algorithms (GA) [14, 15], and can solve the real engineering optimization problems. In 2012, Nakamura et al. [16] introduced the first version of discrete binary bat algorithm (BBA) for feature selection. Ref. [17] proposed another binary version of the bat algorithm (BBA) and used it to deal with the optical buffer design problem.

On the other hand, the application of chaos in optimization algorithms has gained great success in many papers. In Ref. [18], chaotic sequences were used to tune

parameter in genetic algorithm (chaos-genetic algorithm) to solve the scheduling problem. In Ref. [19], the authors introduced chaos into the accelerated particle swarm optimization (CAPSO) to further enhance the global search ability of the algorithm. The success of these combinations shows us that the right use of chaotic maps may improve the performance of the optimization algorithms.

Therefore, we introduce chaos into the BBA, and as a result, propose the chaotic binary bat algorithm (CBBA). Meanwhile, the elitist strategy is applied to the new proposed algorithm to store all the possible global best solutions, which can help to show the advantages of using chaos' non-repetition and ergodicity in the new algorithm. In this paper, the new proposed CBBA is used to deal with the test point selection problem for the first time. Since different chaotic maps may lead to different behavior of the algorithm, we have constructed a set of CBBAs. In these algorithms, we use different chaotic maps to replace the corresponding parameter of BBA. In order to evaluate the proposed algorithms, two analog circuits' examples and a group of statistical experiments are carried out, and the other reported test point selection algorithms are also applied to do the comparison.

Since about 90 % of all the analog faults found in practice are single catastrophic faults [5, 8], we deal with these most commonly encountered single catastrophic faults in analog circuits in this paper. Section 2 presents a brief introduction to BA and BBA. The chaotic maps used in our paper are illustrated in Sect. 3. Section 4 introduces the proposed CBBA and its application in analog test point selection problem. Two analog circuits' examples and a group of statistical experiments are given in Sect. 5. Finally, brief conclusions and future research plan are given in Sect. 6.

The nomenclatures of this paper are as follows:

n_j	The test point j
f_i	The fault i
N	Number of all the test points
N_s	Number of selected test points
M	Number of all the fault states (including fault-free case)
S	Number of separated CUT states
f_i	Frequency of the i -th bat
f_{\min}	Minimum frequency
f_{\max}	Maximum frequency
A	Loudness of emitted sound
r	Pulse emission rate
T_{\max}	Maximum number of iterations

2 Bat algorithm (BA) and binary bat algorithm (BBA)

2.1 Bat algorithm (BA)

The bat algorithm was inspired by the echolocation behavior of bats [14]. In the real world, bats use this type of sonar (echolocation) to detect prey, avoid obstacles, and locate their roosting crevices in the dark. They can emit a very loud sound pulse and use their two ears to listen for the echo that bounces back from the surrounding objects. Bats calculate the time delay from emission and detection of the echo and the loudness variations of the echoes to build up three dimensional scenario of the surrounding. When the target is nearby, their pulse emitting rates increase and the frequency is tuned up. The frequency tuning and the speedup of pulse emission will shorten the wavelength of echolocations and thus increase accuracy of the detection. The BA was developed by using these pivotal factors. The standard BA is based on three idealized rules [14]:

- (1) All bats use echolocation to sense distance, and they also ‘know’ the difference between food/prey and background barriers in some magical way;
- (2) Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{\min} , varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target;
- (3) Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{\min} .

The basic pseudo code of the bat algorithm (BA) is summarized in Fig. 1. The technique is initialized with a swarm of bats collaborate together during the process to find the best solutions. In the BA, each bat (say i) has a defined position x_i and velocity v_i in a d -dimensional search space, and they are updated subsequently during the iterations. Initially, each bat is randomly assigned a frequency which is drawn from interval $[f_{\min}, f_{\max}]$. The new solutions x_i^t and velocities v_i^t at time step t can be calculated by [14]:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{1}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_{Gbest})f_i \tag{2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution, x_{Gbest} is the current global best location (solution) which is located after comparing all the solutions among all the bats.

Bat algorithm

Objective function $f(x)$, $x=(x_1, \dots, x_d)^T$
 Initialize the bat population x_i ($i=1, 2, \dots, n$) and v_i
 Define pulse frequency f_i at x_i
 Initialize pulse rates r_i and the loudness A_i
While ($t < \text{Maximum number of iterations}$)
 Generate new solutions by adjusting frequency
 Update velocities and locations/solutions [equations (1) to (3)]
 if ($\text{rand} > r_i$)
 Select a solution among the best solutions
 Generate a local solution around the selected best solution
 end if
 Generate a new solution by flying randomly
 if ($\text{rand} < A_i$ & $f(x_i) < f(x_{Gbest})$)
 Accept the new solutions
 Increase r_i and reduce A_i
 end if
 Rank the bats and find the current best x_{Gbest}
end while
 Postprocess results and visualization

Fig. 1 Pseudo code of the bat algorithm (BA) [14]

All these equations could guarantee the exploitability of the BA. However, a local search part procedure is also used to perform the exploitation. For this local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk [14]:

$$x_{new} = x_{old} + \epsilon A^t \tag{4}$$

where $\epsilon \in [-1, 1]$ is a random number, A^t is the average loudness of all the bats at this time step.

It is worth pointing out that, to a degree, the BA can be considered as a balanced combination of the standard PSO and the intensive local search controlled by the loudness (A) and pulse rate (r). These two parameters are updated as follows [14]:

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \tag{5}$$

where α and γ are constants. In the simplicity case, we can use $\alpha = \gamma$ [14], and we have in fact used $\alpha = \gamma = 0.9$ in our algorithms. It can be easily found that both the loudness and pulse rate are updated when the new solutions are improved to guarantee that the bats are moving toward the best solutions.

2.2 Binary bat algorithm (BBA)

As each bat moves in the search space towards continuous-valued positions in the bat algorithm, a binary version of the bat algorithm is needed in the test point selection problem. Therefore, the bats’ position should be represented by binary vectors. In Ref. [16], the authors first proposed a binary version of the bat algorithm and used it to deal with the feature selection problem successfully. In their algorithm, they chose the sigmoid function to restrict the bat’s position to binary values. However, the authors in

```

Binary bat algorithm
Initialize the bat population  $x_i$  ( $i=1,2,\dots,n$ )= $\text{rand}(0 \text{ or } 1)$  and  $v_i=0$ 
Define pulse frequency  $f_i$  at  $x_i$ 
Initialize pulse rates  $r_i$  and the loudness  $A_i$ 
While ( $t < \text{Maximum number of iterations}$ )
Adjusting frequency and updating velocities
Calculate transfer function value using equation (6)
Update locations/solutions using equations (7)
  if ( $\text{rand} > r_i$ )
    Select a solution ( $x_{Gbest}$ ) among the best solutions
    Change some of the dimensions of position vector with some of
    the dimensions of  $x_{Gbest}$ 
  end if
  Generate a new solution by flying randomly
  if ( $\text{rand} < A_i$  &  $f(x_i) < f(x_{Gbest})$ )
    Accept the new solutions
    Increase  $r_i$  and reduce  $A_i$ 
  end if
Rank the bats and find the current best  $x_{Gbest}$ 
end while
    
```

Fig. 2 Pseudo code of the binary bat algorithm (BBA) [17]

Ref. [17] illustrated that their proposed v-shaped transfer function was better than the sigmoid function in the binary version of the bat algorithm, and used it to deal with the optical buffer design problem successfully. We chose to use their better strategy in our algorithm. The v-shaped transfer function and the position updating rule are listed as follows [17]:

$$V(v_{ik}^{t+1}) = \left| \frac{2}{\pi} \arctan\left(\frac{2}{\pi} v_{ik}^t\right) \right| \tag{6}$$

$$x_{ik}^{t+1} = \begin{cases} (x_{ik}^t)^{-1} & \text{If } \text{rand} < V(v_{ik}^{t+1}) \\ x_{ik}^t & \text{rand} \geq V(v_{ik}^{t+1}) \end{cases} \tag{7}$$

where v_{ik}^t and x_{ik}^t indicate the velocity and position of the i -th bat at iteration t in the k -th dimension, and $(x_{ik}^t)^{-1}$ is the complement of x_{ik}^t .

The general pseudo code of the BBA is presented in Fig. 2. The changes from the standard BA are underlined.

3 Chaotic maps

Chaotic optimization method is a kind of random-based optimization method, which uses chaotic variables instead of random variables. Due to the non-repetition and ergodicity of chaos, it can carry out overall searches at higher speeds than stochastic searches that depend on probabilities [20]. Chaos is a deterministic and random-like process, which is very sensitive to its initial conditions and parameters. The nature of chaos is apparently random and unpredictable, but it also possesses an element of regularity [21].

In almost all the metaheuristic algorithms with stochastic components, randomness is achieved by using

some probability distributions (such as uniform). In principle, replacing such randomness by chaotic maps could obtain better algorithms since chaos can have very similar properties of randomness with better statistical and dynamical properties. This can help to reach nearly all the different modes and ensure the diversity of the solutions generated by the algorithm. The chaotic search can escape from local optima more easily compared with other stochastic search optimization algorithms. Therefore, many researchers [18–21] have combined chaos with optimization algorithms to enhance the search efficiency and prevent from premature convergence to local minima.

To enhance the diversity of solution and improve the effectiveness and robustness of the BBA, one-dimensional, non-invertible maps are used to generate chaotic sequences in our proposed CBBA. Here we outline some well known chaotic maps which will be used for later simulations.

- (1) Chebyshev map
The family of Chebyshev map [22, 23] is defined as follows:

$$x_{i+1} = \cos(i \cdot \cos^{-1}(x_i)) \tag{8}$$

- (2) Tent map
Tent map [22, 24] is defined by the following iteration equation:

$$x_{i+1} = \begin{cases} x_i/0.7 & x_i < 0.7 \\ 10(1 - x_i)/3 & x_i \geq 0.7 \end{cases} \tag{9}$$

- (3) Circle map
Circle map [22, 25] is represented by the following equation:

$$x_{i+1} = x_i + b - \left(\frac{a}{2\pi}\right) \sin(2\pi x_i) \text{mod}(1) \tag{10}$$

where $a = 0.5$ and $b = 0.2$

- (4) Iterative map
The iterative chaotic map with infinite collapses [22, 26] can be defined as:

$$x_{i+1} = \sin\left(\frac{a\pi}{x_i}\right) \tag{11}$$

where $a \in (0, 1)$ is a suitable parameter. We set $a = 0.7$ in our later experiments.

- (5) Logistic map
Logistic map [22, 26] can be written as:

$$x_{i+1} = ax_i(1 - x_i) \tag{12}$$

where $x_0 \in (0, 1), x_0 \notin \{0.00, 0.25, 0.50, 0.75, 1.00\}$. We set $a = 4.0$ in our later experiments.

- (6) Piecewise map
Piecewise map [22, 27] can be described as:

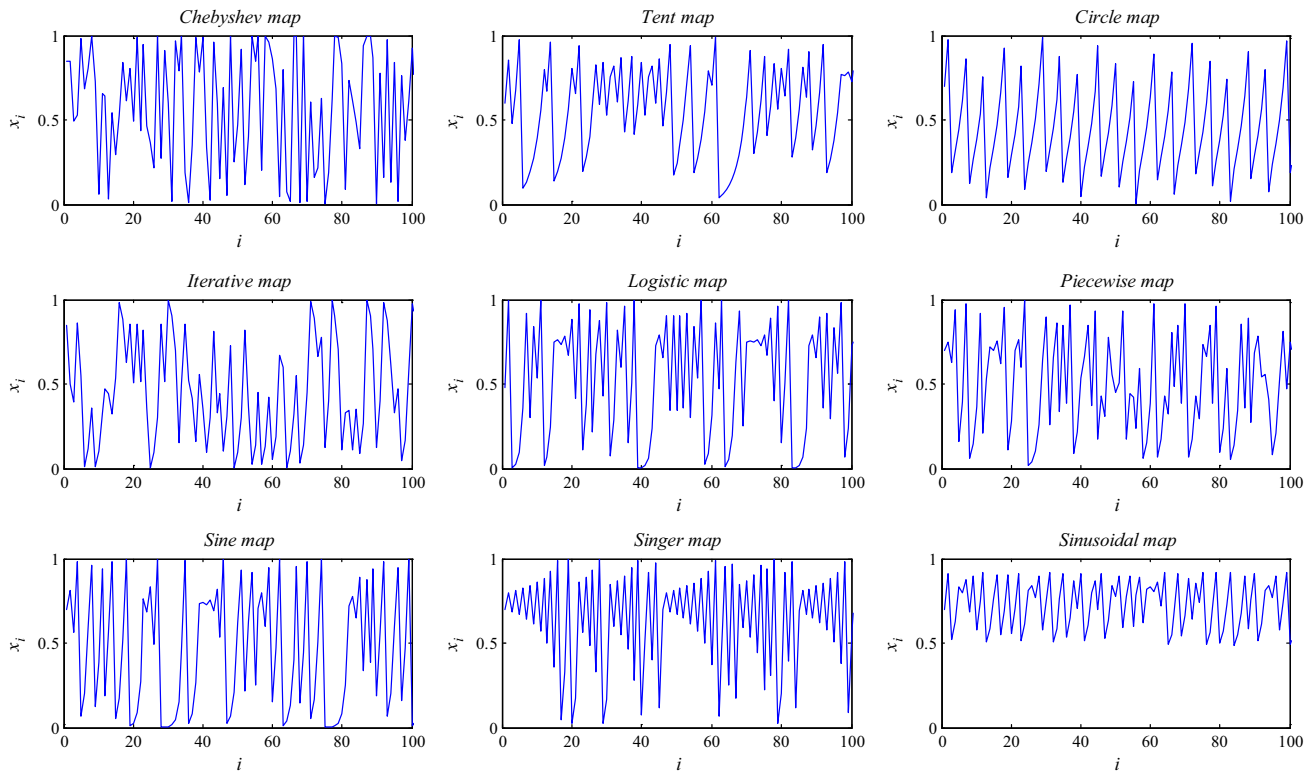


Fig. 3 Visualization of chaotic maps

$$x_{i+1} = \begin{cases} \frac{x_i}{p} & 0 \leq x_i < p \\ \frac{x_i - p}{0.5 - p} & p \leq x_i < 0.5 \\ \frac{1 - p - x_i}{0.5 - p} & 0.5 \leq x_i < 1 - p \\ \frac{1 - x_i}{p} & 1 - p \leq x_i < 1 \end{cases} \quad (13)$$

where $p \in (0, 0.5)$, and we set $p = 0.4$ in our later experiments.

(7) Sine map

Sine map [22, 28] can be described as:

$$x_{i+1} = \frac{a}{4} \sin(\pi x_i) \quad (14)$$

where $0 < a \leq 4$, and we set $a = 4$ in our later experiments.

(8) Singer map

One dimensional singer map [22, 29] is formulated as:

$$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.3x_i^4) \quad (15)$$

where $\mu \in (0.9, 1.08)$, and we set $\mu = 1.07$ in our later experiments.

(9) Sinusoidal map

Sinusoidal map [19, 22] is generated as the following equation:

$$x_{i+1} = ax_i^2 \sin(\pi x_i) \quad (16)$$

where $a = 2.3$.

Figure 3 shows the chaotic value distributions of 100 iterations for all the above listed maps with random initial values. It should be noted that the chaotic maps which do not generate values between 0 and 1 are normalized to have the same scale.

4 New proposed CBBA and its application in analog test point selection

In order to enhance the diversity of solution and meanwhile improve the effectiveness and robustness of the BBA, the chaotic maps are used to tune the BBA parameter and the elitist strategy is also introduced to store all the possible global best solutions, which will lead to a set of CBBAs. The construction of the CBBA and its application in analog test point selection will be discussed in this section.

4.1 Construction of the CBBA

4.1.1 Using of chaotic maps

In our proposed CBBA, parameter β of Eq. (1) is modified by chaotic maps during iterations, and the new frequency equation is written as:

$$f_i = f_{\min} + (f_{\max} - f_{\min})c_i \quad (17)$$

where c_i is the chaotic map value in the i -th iteration. It should be noted that all the chaotic maps here are normalized between 0 and 1. Since the chaotic maps are sensitive to the initial conditions and parameters, we should decide the proper parameter values by experiments. In order to simplify the procedure, meanwhile, do the comparison conveniently, we chose the same initial parameter values of chaotic maps as Ref. [22] in our experiments.

Since nine different chaotic maps are listed in Sect. 3, we could obtain nine different CBBA's totally.

4.1.2 Elitist strategy

It is possible that the global best solution may be more than one in reality [12]. Therefore, all the best solutions should be stored during the non-repetition and ergodicity searching of the algorithm. An elitist set XG_{best} is defined in the proposed CBBA to get more possible solutions. If more than one global best solutions (have got the same best fitness value) is obtained during the iteration, they can synchronously be stored in the elitist set. In the next iteration, if there are more new global best solutions that have the same fitness value as these in the elitist set, these new solutions are add into the elitist set. If these new global best solutions have better fitness values, all the old solutions in the elitist set are replaced by these new best solutions. Finally, we will get all the optimal solutions with the same best fitness value in the elitist set.

The general pseudo code of the CBBA is presented in Fig. 4. The changes from the BBA are underlined.

4.2 Application of CBBA in analog test point selection

As discussed in Sect. 1, the integer-coded dictionary has been proven to be an effective tool for optimal selection of test points; the utilization of CBBA in analog test point selection is also based on the integer-coded dictionary of the analog circuit. Besides the construction of integer-coded fault dictionary, inoculating CBBA with the test point selection problem rightly is also very important. We will discuss the details about them below.

Chaotic binary bat algorithm

```

Initialize the bat position  $x_i = \{x_{ik} \mid (i=1, 2, \dots, n; k=1, 2, \dots, d) = \text{rand}(0 \text{ or } 1)\}$  and  $v_i = 0$ 
Initialize the elitist set  $XG_{best}$  as a null
Define pulse frequency  $f_i$  at  $x_i$ 
Initialize pulse rates  $r_i$  and the loudness  $A_i$ 
While ( $t < \text{Maximum number of iterations } T_{max}$ )
  Use one of the chaotic map generate equations [equations (8) to (16)] to
  obtain the chaotic sequence value  $c_i$ 
  Calculate the new frequency value using equation (17)
  Calculate transfer function value using equation (6)
  Update locations/solutions using equations (7)
  if ( $\text{rand} > r_i$ )
    Select a solution ( $x_{G_{best}}$ ) among the best solutions
    Change some of the dimensions of position vector with some of
    the dimensions of  $x_{G_{best}}$ 
  end if
  Generate a new solution by flying randomly
  if ( $\text{rand} < A_i$  &  $f(x_i) < f(x_{G_{best}})$ )
    Accept the new solutions
    Increase  $r_i$  and reduce  $A_i$ 
  end if
  Rank the bats and find the current best  $x_{G_{best}}$ 
  Update the elitist set  $XG_{best}$ 
  If (the new best solutions have the same best fitness value as the last iteration
  and the new best solutions are not in the elitist set  $XG_{best}$ )
    Add the new solutions to the elitist set  $XG_{best}$ 
  end if
  If (the new best solutions have better fitness value)
    Replace the solutions in the elitist set  $XG_{best}$  with the new solutions
  end if
end while

```

Fig. 4 Pseudo code of the chaotic binary bat algorithm (CBBA)

4.2.1 Integer-coded fault dictionary of analog circuits

Hochwald and Bastian [8] first proposed the concept of ambiguity sets and defined a diode drop (0.7 V) as the ambiguity gap. The ambiguity group is defined as that any two faulty cases fall into the same ambiguity set if the gap between the voltage values of their responses is less than 0.7 V. Lin and Elcherif [3] first proposed the integer-coded fault dictionary technique in 1985. The fault dictionary provides information about the ambiguity sets for each test point and the mutual information among them [10]. The purpose of test point selection is to separate all the faults (including fault-free case) with a minimum number of test points, and reduce the dimension of the fault dictionary.

The integer-coded fault dictionary is a two dimensional integer matrix. Its rows represent all the faults (including fault-free case), and its columns show all candidate test points. For each fault, an integer code is generated from the numbers of ambiguity sets of each test point, and the same integer number represents all the faults that belong to the same ambiguity group in a given candidate test point. Since each candidate test point represents an independent measurement, the ambiguity groups of different test points can be numbered using the same integer without confusion [5, 8]. More details about fault dictionary and ambiguity set can be found in Ref. [3, 5].

4.2.2 Initialize parameters of CBBA

The best solutions are the bats with the best position that CBBA can explore. In this paper, the position of each bat is

initialized as the binary vector. The number of elements in this binary vector is the same as the number of test points N . The position of the i -th bat is defined by

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iN}) \tag{18}$$

where $x_{ij} = 1$ or 0 , which indicates whether the corresponding test point is selected or not. That is to say, if the j -th test point is selected by CBBA, $x_{ij} = 1$; otherwise, $x_{ij} = 0$.

The bats' velocity will decide the position of the bats in the next iteration. For simplicity and comparison, the velocity of the bat is initialized as a zero vector (the same as Ref. [17]), which has the same dimension as the position vector.

4.2.3 Fitness function

For the analog test point selection problem, we hope to separate all the faults with minimum test points. To evaluate the quality of the i -th solution properly, the fitness function is defined as:

$$Fitness_i = \frac{S_i + (N - Ns_i)}{M + N} \tag{19}$$

where S_i is the number of separated CUT states, Ns_i is the number of selected test points, M is the number of all the fault states (including fault-free case), N is the number of all the test points.

It can be seen from the fitness function that the fitness value is in the region $[0, 1]$ and the better solution always has larger fitness value. Therefore, the best solutions will get the largest fitness value during the iterations.

4.2.4 Work principle of CBBA for test point selection

As discussed above, the bats' positions in CBBA denote the results of the test point selection. After initialization, all the bats will have a random position. And then, the algorithm goes into the iteration phase. The fitness function [Eq. (19)] is used to calculate the fitness values of all the bats. The algorithm will find out all the bats with largest fitness values during the iteration, and judge whether their locations should be added into the elitist set XG_{best} (or even replace all the solutions in the elitist set sometimes). The loudness (A_i), pulse rate (r_i), chaotic map value (c_i), frequency (f_i), velocity (v_i) and position (x_i) of each bat will be updated during the iterations. The algorithm is going to be finished at the end of the maximal iteration T_{max} , which should be set as a proper number according to the complexity of the problem. We will finally get all the optimum solutions in the elitist set XG_{best} .

4.2.5 Time complexity of CBBA for test point selection

For analog test point selection with M fault states and N test points, there exist T_{max} iterations for the CBBA. Based on the fitness function [Eq. (19)], when evaluating each solution (bat's position), we need to check the number of isolated fault states. This can be transformed into a sorting problem and the computational time complexity is $O(M \log M)$. If the population size of bats is P , we need to evaluate P solutions for each iteration of the algorithm and the time complexity is $O(PM \log M)$. Therefore, the total time complexity of CBBA is $O(T_{max}PM \log M)$.

5 Experiments

In order to demonstrate the feasibility and effectiveness of the CBBA for test point selection, meanwhile, compare with other reported methods based on the integer-coded dictionary, we have done different kinds of experiments (two analog circuits' examples and a group of statistical experiments). Since nine well known chaotic maps have been introduced in Sect. 3 and different chaotic maps used in CBBA may lead to different results, all these chaotic maps are adopted to construct the CBBA respectively in our experiments. The names of these methods (CBBAs) and the corresponding chaotic maps are listed in Table 1.

During all the experiments, the population size of bats is set as 40, the dimensions of the position and velocity vectors are set as N (the number of test points). For comparison and simplicity, we referred to the Ref. [17] and set the initial loudness (A_0) as 0.25, the initial pulse rate (r_0) as 0.2. The maximum number of iterations T_{max} is set as 100. The initial parameters of chaotic maps are set as the discussions in Sects. 3 and 4.1.1. By the way, all the parameters can be modulated experimentally in practice to suit different applications.

In order to highlight the improvements of our proposed CBBA, the BBA in Ref. [17] is also used to do all the

Table 1 Methods and the corresponding chaotic maps

Methods	Chaotic maps adopted in the method
CBBA1	Chebyshev map
CBBA2	Tent map
CBBA3	Circle map
CBBA4	Iterative map
CBBA5	Logistic map
CBBA6	Piecewise map
CBBA7	Sine map
CBBA8	Singer map
CBBA9	Sinusoidal map

experiments. The relevant parameters of BBA are set the same as the CBBA.

All the programs run on an Intel 3.2 GHz processor computer with MATLAB R2013a circumstance during the experiments.

5.1 Video amplifier example

The experiment is implemented on a video amplifier circuit that is shown in Fig. 5. This circuit is also cited by Ref. [12] to verify the efficiency of the MDFDPSO method. The excitation signal is a 5-kHz 8-V sinusoidal wave. The marked voltage sources “V1” and “V2” in Fig. 5 are 5-kHz -6-V and 5-kHz 6-V sinusoidal waves. Totally, there are sixteen potential catastrophic faults f_1 to f_{16} (including the nominal case, $M = 16$) and thirteen test points n_1 to n_{13} ($N = 13$). The responses of voltage values at all test points for different faulty conditions are obtained by PSPICE simulation. The integer-coded fault dictionary is constructed and shown in Table 2 (it's the same as the fault dictionary obtained in Ref. [12]).

After running, all the proposed nine different CBBA (CBBA1–CBBA9) have found the same best solutions in their elitist set:

$$XG_{best} = \begin{cases} n_2, n_7, n_8, n_9, n_{12} \\ n_3, n_7, n_8, n_9, n_{12} \\ n_7, n_8, n_9, n_{10}, n_{12} \end{cases}.$$

The final best solutions are the same as the results in Ref. [12]. For comparison, the GA [11], MDFDPSO [12], EI [4], HGS [5] and BBA [17] are applied in this example. For each algorithm, 100 independent runs are implemented. The results are shown in Table 3.

As indicated in Table 3, all the algorithms achieve 100 % of success. Although EI and HGS cost shorter computation time, they can only find one best solution while other methods (except GA and BBA) can find three best solutions. Comparing with GA and MDFDPSO method, our proposed algorithms (CBBA1–CBBA9) cost less computation time and have better efficiency. The improvement of our methods (compare with BBA) is obvious (nearly the same computation time but more best solutions).

Finally, it comes to a conclusion that our new proposed algorithms (CBBA) are superior to other mentioned methods.

5.2 Leapfrog filter amplifier example

The experiment is implemented on a leapfrog filter circuit that is shown in Fig. 6. This circuit is also cited by Ref. [5] to verify the efficiency of the HGS method. The excitation signal is a 1-kHz 4-V sinusoidal wave. The low-input-impedance failure of the amplifiers and some resistance and capacitance failures are considered in this circuit. Totally, there are twenty-three potential catastrophic faults

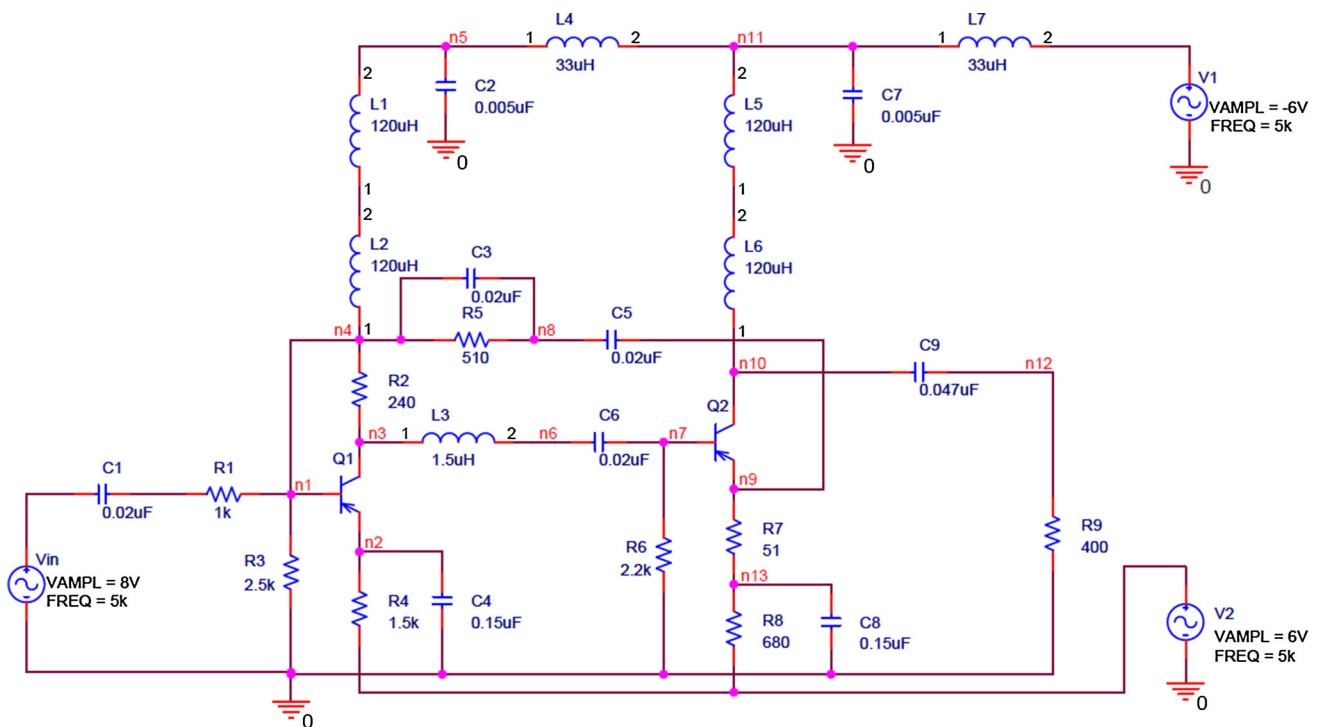


Fig. 5 Video amplifier with two transistors

Table 2 Integer-coded fault dictionary for video amplifier

Faults	N ₁	N ₂	N ₃	n ₄	n ₅	n ₆	n ₇	n ₈	n ₉	n ₁₀	n ₁₁	n ₁₂	n ₁₃
f ₁ (NOM)	1	1	1	4	2	1	5	2	4	3	1	3	2
f ₂ (L1/L2 open)	2	1	2	3	2	2	4	3	4	3	1	3	2
f ₃ (L4 open)	3	1	4	2	1	3	3	4	4	3	1	3	2
f ₄ (L3 open)	1	1	1	4	2	2	3	2	4	3	1	3	2
f ₅ (L7 open)	3	1	4	2	1	3	3	4	4	1	2	2	2
f ₆ (L5/L6 open)	1	1	1	4	2	1	5	2	4	2	1	2	2
f ₇ (Q1 B open)	1	4	3	4	2	1	5	2	4	5	1	3	2
f ₈ (Q2 B open)	1	1	1	4	2	1	5	2	5	4	1	3	2
f ₉ (C4 short)	1	1	1	4	2	1	5	4	4	3	1	3	2
f ₁₀ (C6 short)	1	2	5	4	2	1	5	2	4	4	1	3	2
f ₁₁ (C3 short)	1	1	1	4	2	1	5	3	1	3	1	3	3
f ₁₂ (C5 short)	1	1	1	4	2	1	1	2	4	3	1	3	2
f ₁₃ (C8 short)	1	1	1	4	2	1	5	2	2	4	1	3	1
f ₁₄ (C9 short)	1	1	1	4	2	1	5	2	4	3	1	1	2
f ₁₅ (Q1 BE short)	4	3	3	1	2	4	2	1	4	4	1	3	2
f ₁₆ (Q2 BE short)	1	1	1	4	2	1	6	2	3	4	1	3	2

Table 3 Comparison of CBBA with other methods for video amplifier

Methods	Percentage of success (%)	Average running time (in seconds)	Number of best solutions
GA	100	4.858	1
MDFDPSO	100	0.981	3
EI	100	0.023	1
HGS	100	0.019	1
BBA	100	0.550	1
CBBA1	100	0.567	3
CBBA2	100	0.573	3
CBBA3	100	0.574	3
CBBA4	100	0.567	3
CBBA5	100	0.571	3
CBBA6	100	0.572	3
CBBA7	100	0.583	3
CBBA8	100	0.566	3
CBBA9	100	0.568	3

f₁ to f₂₃ (including the nominal case, $M = 23$) and twelve test points n_1 to n_{12} ($N = 12$). The responses of voltage values at all test points for different faulty conditions are obtained by PSPICE simulation. The integer-coded fault dictionary is constructed and shown in Table 4 (it's the same as the fault dictionary obtained in Ref. [5]).

After running, all the proposed nine different CBBAs (CBBA1 ~ CBBA9) have found the same best solutions in their elitist set:

$$XG_{best} = \left\{ \begin{matrix} n_1, n_2, n_3, n_4, n_7, n_8, n_{11} \\ n_1, n_2, n_3, n_4, n_7, n_9, n_{11} \\ n_1, n_2, n_3, n_4, n_7, n_{10}, n_{11} \end{matrix} \right\}.$$

One of the final best solutions is the same as the results in Ref. [5]. For comparison, the GA [11], MDFDPSO [12], EI [4], HGS [5] and BBA [17] are also applied in this example. For each algorithm, 100 independent runs are implemented. The results are shown in Table 5.

As indicated in Table 5, all the algorithms achieve 100 % of success except the MDFDPSO method (99 %). EI and HGS cost shorter computation time, but they can only find one best solution while other methods (except GA and BBA) can find three best solutions. Comparing with GA and MDFDPSO method, our proposed algorithms (CBBA1 ~ CBBA9) cost less computation time and have

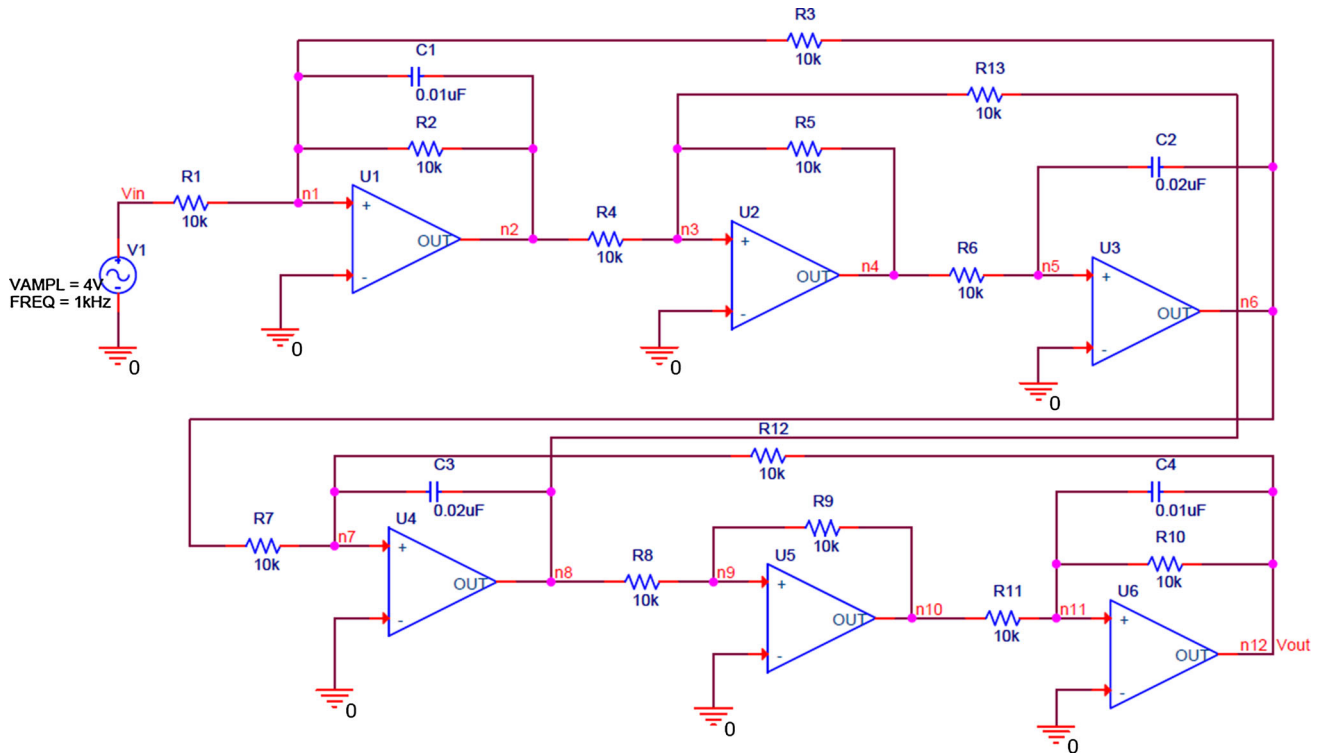


Fig. 6 Leapfrog filter

Table 4 Integer-coded fault dictionary for leapfrog filter

Faults	n ₁	n ₂	n ₃	n ₄	n ₅	N ₆	N ₇	N ₈	N ₉	N ₁₀	N ₁₁	n ₁₂
f ₁ (NOM)	4	2	2	3	1	5	3	3	2	3	3	3
f ₂ (C2 open)	8	7	6	7	4	9	7	9	4	6	4	7
f ₃ (C2 short)	4	3	2	3	1	3	3	2	2	2	3	2
f ₄ (C3 open)	1	0	0	0	0	1	1	1	1	1	1	1
f ₅ (R1 open)	9	8	6	7	4	10	7	9	4	6	4	7
f ₆ (R1 short)	5	5	4	6	4	8	7	8	4	6	4	7
f ₇ (R2 open)	6	7	6	6	4	9	7	8	4	6	4	7
f ₈ (R2 short)	4	1	2	1	1	3	3	2	2	2	3	2
f ₉ (R3 open)	2	0	0	0	0	0	0	0	0	0	0	0
f ₁₀ (R3 short)	0	0	0	0	0	1	1	0	0	0	0	0
f ₁₁ (R4 open)	4	3	2	1	1	3	3	2	2	2	3	2
f ₁₂ (R7 short)	1	0	0	0	0	1	1	0	0	0	0	0
f ₁₃ (R8 open)	8	7	5	6	4	9	6	6	4	6	4	6
f ₁₄ (R10 open)	4	4	2	2	1	4	3	3	2	3	3	5
f ₁₅ (R11 open)	7	6	3	5	3	7	4	7	4	6	4	7
f ₁₆ (R12 short)	1	0	0	0	0	1	0	0	0	0	0	0
f ₁₇ (R13 open)	4	4	2	4	1	6	3	4	2	4	3	4
f ₁₈ (U1 fail)	3	0	0	0	0	1	1	0	0	0	0	0
f ₁₉ (U2 fail)	1	0	1	0	0	1	1	1	1	1	1	1
f ₂₀ (U3 fail)	8	7	6	6	2	9	7	8	4	6	4	7
f ₂₁ (U4 fail)	1	0	0	0	0	1	2	1	1	1	1	1
f ₂₂ (U5 fail)	8	7	6	7	4	9	7	7	3	5	4	7
f ₂₃ (U6 fail)	1	0	0	0	0	1	1	1	1	1	2	1

Table 5 Comparison of CBBA with other methods for leapfrog filter

Methods	Percentage of success (%)	Average running time (in seconds)	Number of best solutions
GA	100	5.521	1
MDFDPSO	99	1.130	3
EI	100	0.025	1
HGS	100	0.022	1
BBA	100	0.620	1
CBBA1	100	0.638	3
CBBA2	100	0.642	3
CBBA3	100	0.643	3
CBBA4	100	0.639	3
CBBA5	100	0.640	3
CBBA6	100	0.643	3
CBBA7	100	0.651	3
CBBA8	100	0.635	3
CBBA9	100	0.637	3

better efficiency and accuracy. The improvement of our methods (compare with BBA) is obvious (nearly the same computation time but more best solutions).

Therefore, we can draw the same conclusion with the video amplifier circuit example that our new proposed algorithms (CBBAs) are superior to other mentioned methods in dealing with the test point selection problem because of their higher efficiency, better accuracy and robustness.

5.3 Statistical experiments

Although the above experiments have shown the great advantage and ability of the proposed algorithms in finding the optimum test point sets, there still no theoretical proof can be offered to demonstrate a specific non-exhaustive algorithm's optimality so far. The new proposed algorithms (CBBAs) must be statistically tested on larger number of fault dictionaries to demonstrate their efficiency and qualities of generating optimum test point sets.

Such statistical experiments are carried out on the randomly computer-generated fault dictionaries and the final solutions are found by using the GA [11], MDFDPSO [12], EI [4], HGS [5], BBA [17] and the proposed methods (CBBA1–CBBA9) respectively. Here, a total of 100 integer-coded dictionaries are randomly created. Each dictionary is generated for a hypothetical CUT with 100 simulated faults, 25 test points and randomly five ambiguity sets per test point. Meanwhile, all the randomly created dictionaries are first analyzed by the exhaustive search algorithm to make sure that all the 100 dictionaries have five test points in their best solutions (optimum sets). The final results are shown in Tables 6 and 7.

As can be seen from Tables 6 and 7, the proposed methods (CBBA1–CBBA9), together with GA and BBA method, can always find the optimum solutions (success

rate is 100 %), meanwhile, the MDFDPSO method achieves 96 % of success (with 4 % chances to get trapped into local optima), but our proposed methods (CBBA1–CBBA9) can find more final best solutions. Although the time consumptions of the EI and HGS method are radically lower than the other methods, their final solution accuracy rates are lower. Therefore, through statistic experiments, the proposed methods (CBBAs) also reveal their high accuracy and effectiveness and are demonstrated to be superior to other methods.

Actually, whether the data in the integer-coded fault dictionary are randomly computer generated or derived from a realistic circuit, it does not spoil the validity of the proposed methods (CBBAs).

5.4 Practical application analysis

Although the components of the analog circuit with parameter tolerance may be encountered in practice, it doesn't affect the efficiency of our proposed algorithms either. If only the integer-coded fault dictionary can be constructed properly, our proposed algorithms (CBBAs) can be used to find the optimum test point sets.

From the experiments carried out above, we may come to the conclusion that our new proposed CBBAs can find nearly all the best solutions efficiently. However, which best solution should be used during our practice becomes a problem. Since each test point in the actual circuit has a test cost, which can be defined as the manpower requirements, equipment requirements, measure time and technique requirements, test complexity or even other economic factors and so on. Therefore, the test cost of each candidate test point is a constant value for an actual analog circuit in practice and can be estimated. Since minimizing the total expected test cost is an important target in practice, the best

Table 6 Statistical results of the solution accuracy

Size of the optimum sets found	Percentage of the optimum sets with a specific size found by a specific method and the average number of best solutions													
	GA	MDFPSO	EI	HGS	BBA	CBBA1	CBBA2	CBBA3	CBBA4	CBBA5	CBBA6	CBBA7	CBBA8	CBBA9
5	100	96	68	49	100	100	100	100	100	100	100	100	100	100
6	0	4	32	51	0	0	0	0	0	0	0	0	0	0
Average number of best solutions	1	4.15	1	1	1	12.96	11.84	12.04	13.02	11.92	12.87	12.22	12.54	13.05

Table 7 The average running time of each method

The average execution time per dictionary for the experimented methods (in seconds)													
GA	MDFPSO	EI	HGS	BBA	CBBA1	CBBA2	CBBA3	CBBA4	CBBA5	CBBA6	CBBA7	CBBA8	CBBA9
12.360	2.649	0.061	0.064	1.375	1.384	1.393	1.383	1.376	1.381	1.380	1.412	1.385	1.379

solution with the lowest test cost of all the best solutions would be the real best solution in practice. Therefore, our CBBAs have significant practical application value.

6 Conclusion and future research

The analog test point selection problem remains an important research field in fault diagnosis of analog circuit. How to find a minimum set of test points efficiently to isolate all the faults to a desired degree, therefore, becomes the key point. In the present work, we have introduced chaos and elitist strategy to the BBA and developed a set of CBBAs, meanwhile, for the first time applied them to deal with the analog test point selection problem. The attractiveness of our proposed CBBA mainly lies in two aspects: the utilization of different chaotic maps to tune the BBA parameter and the application of elitist strategy to store all the possible global best solutions. These improvements obviously enhance the performance of BBA and make our new algorithms (CBBAs) obtain all the best solutions (usually more than one) easily and effectively, which will give us more possible choices in the real engineering practice. Carried out on the same trademark analog circuits, the proposed algorithms show great advantage in finding the best solutions than the other mentioned methods. Since no theoretical proof can be given to the optimality of the proposed algorithms, statistical experiments are utilized for their evaluation. The results demonstrate that our proposed methods (CBBAs) have higher efficiency, better accuracy and quality in finding the optimum test point sets. They are effective and feasible methods in minimizing the size of the test point set of the analog circuit, and can help to find the best solution with lowest test cost of all in practice. Therefore, they are particularly applicable to actual circuits and engineering practice.

CBBA offers new techniques for improving the properties of BBA and whether it is fit for solving other optimization problems is being investigated. In addition, we used default values referred to other researchers' work for the variable parameters of some chaotic maps in this paper, and how these parameters affect the solutions so as to further improve the performance of CBBA will be our future research direction.

References

- Duhamal, P., & Rault, J. C. (1979). Automatic tests generation techniques for analog circuits and systems: A review. *IEEE Transactions on Circuits and Systems I*, 26, 411–440.
- Bandler, J. W., & Salama, A. E. (1985). Fault diagnosis of analog circuits. *Proceedings of the IEEE*, 73, 1279–1325.
- Lin, P. M., & Elcherif, Y. S. (1985). Analogue circuits fault dictionary—new approaches and implementation. *International Journal of Circuit Theory and Applications*, 13(2), 149–172.
- Starzyk, J. A., Liu, Dong, Liu, Zhi-Hong, Nelson, D. E., & Rutkowski, J. O. (2004). Entropy-based optimum test nodes selection for analog fault dictionary techniques. *IEEE Transactions on Instrumentation and Measurement*, 53, 754–761.
- Yang, C. L., Tian, S. L., & Long, B. (2009). Application of heuristic graph search to test point selection for analog fault dictionary techniques. *IEEE Transactions on Instrumentation and Measurement*, 58(7), 2145–2158.
- Prasad, V. C., & Babu, N. S. C. (2000). Selection of test nodes for analog fault diagnosis in dictionary approach. *IEEE Transactions on Instrumentation and Measurement*, 49(6), 1289–1297.
- Varghese, X., Williams, J. H., & Towill, D. R. (1978). Computer aided feature selection for enhanced analogue system fault location. *Pattern Recognition*, 10(4), 265–280.
- Hochwald, W., & Bastian, J. D. (1979). A dc approach for analog fault dictionary determination. *IEEE Transactions on Circuits and Systems*, 26, 523–529.
- Spaandonk, J., & Kevenaar, T. (1996). Iterative test point selection for analog circuits. In *Proc. of the 14th VLSI test symp.*, Princeton, NJ, USA (pp. 66–71).
- Pinjala, K. K., Bruce, C. K. (2003). An approach for selection of test points for analog fault diagnosis. In *Proceedings of the 18th IEEE international symposium on defect and fault tolerance in VLSI systems* (pp. 287–294).
- Golonek, T., & Rutkowski, J. (2007). Genetic-algorithm-based method for optimal analog test points selection. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 54(2), 117–121.
- Jiang, R. H., Wang, H. J., Tian, S. L., & Long, B. (2010). Multidimensional fitness function DPSO algorithm for analog test point selection. *IEEE Transactions on Instrumentation and Measurement*, 59(6), 1634–1641.
- Luo, H., Wang, Y., Lin, H., & Jiang, Y. (2012). A new optimal test node selection method for analog circuit. *Journal of Electronic Testing*, 28(3), 279–290.
- Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In J. R. Gonzalez, et al. (Eds.), *Nature inspired cooperative strategies for optimization (NICSO 2010)* (Vol. 284, pp. 65–74). Berlin: Springer.
- Yang, X. S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computation*, 29(5), 464–483.
- Nakamura, R. Y. M., Pereira, L. A. M., Costa, K. A., Rodrigues, D., Papa, J. P. & Yang, X. S. (2012) BBA: A binary bat algorithm for feature selection. In *25th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, IEEE Publication (pp. 291–297).
- Mirjalili, S., Mirjalili, S. M., & Yang, X. S. (2014). Binary bat algorithm. *Neural Computing & Applications*, 25, 663–681.
- Gharooni-fard, G., Moein-darbari, F., Deldari, H., & Morvaridi, A. (2010). Scheduling of scientific workflows using a chaos-genetic algorithm. *Procedia Computer Science*, 1, 1445–1454.
- Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 18(2), 327–340.
- Coelho, L., & Mariani, V. C. (2008). Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Systems with Applications*, 34, 1905–1913.
- Alatas, B., Akin, E., & Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals*, 40, 1715–1734.
- Saremi, S., Mirjalili, S., & Lewis, A. (2014). Biogeography-based optimisation with chaos. *Neural Computing & Applications*, 25, 1077–1097.

23. Tavazoei, M. S., & Haeri, M. (2007). Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Applied Mathematics and Computation*, 187, 1076–1085.
24. Ott, E. (2002). *Chaos in dynamical systems*. Cambridge, UK: Cambridge University Press.
25. Hilborn, R. C. (2004). *Chaos and nonlinear dynamics: An introduction for scientists and engineers* (2nd ed.). New York: Oxford University Press.
26. May, Robert M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261, 459–467.
27. Li, Y. T., Deng, S. J., & Xiao, D. (2011). A novel Hash algorithm construction based on chaotic neural network. *Neural Computing and Applications*, 20, 133–141.
28. Devaney, R. L. (1987). *An introduction to chaotic dynamical systems*. Redwood City, CA: Addison-Wesley.
29. Peitgen, H., Jurgens, H., & Saupe, D. (1992). *Chaos and fractals*. Berlin: Springer.



Dongsheng Zhao was born in 1981. He received the B.S. degree in electronic science and technology from Beijing Jiaotong University, Beijing, China, in 2006 and the M.S. degree in precision instruments and machinery from Xi'an Shiyou University, Xi'an, China, in 2012. He is currently working toward the Ph.D. degree at the School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing, China. His research

interests include analog circuit test and measurement, electronic

testing and fault diagnosis of complicated electronic systems, signal processing, artificial intelligence and pattern recognition.



Yuzhu He was born in 1955. He received the B.S. degree from Beihang University, Beijing, China, in 1994 and the Ph.D. degree from City University of Hong Kong, Hong Kong, China, in 2004. He is currently a professor and Ph.D. supervisor in School of Instrumentation Science and Opto-electronics Engineering, Beihang University. His teaching and researching interests include modern electronic system testing and testing technology, universal test system

integration, electronic system fault diagnosis and location technology.