
Parameter identification for discrete memristive chaotic map using adaptive differential evolution algorithm

Yuexi Peng · Shaobo He* · Kehui Sun

Received: date / Accepted: date

Abstract Since the concept of discrete memristor was proposed, more and more scholars began to study this topic. At present, most of the works on the discrete memristor are devoted to the mathematical modeling and circuit implementation, but the research on its synchronization control has not received much attention. This paper focuses on the parameter identification for the discrete memristive chaotic map, and a modified intelligent optimization algorithm named adaptive differential evolution algorithm is proposed. To deal with the complex behaviors of hyperchaos and coexisting attractors of the considered discrete memristive chaotic maps, the identification objective function adopts two special parts: time sequences and return maps. Numerical simulations demonstrate that the proposed algorithm has the best performance among the six existing algorithms, and it can still accurately identify the parameters of the original system under noise interference.

Keywords Parameter identification · discrete memristor · chaotic map · differential evolution algorithm · return maps · coexisting attractors

1 Introduction

Memristor is a kind of nonlinear electronic component which connects the relationship between magnetic flux and electric charge in the circuit [1]. Since Strukov et al. [2] successfully designed the real memristor in 2008, the research on memristor becomes a hot topic. And it has received extensive attentions in the fields of materials [3], artificial intelligence [4, 5], circuits [6, 7], etc.

The memristor is completely different from the traditional nonlinear electronic devices. Due to its special nonlinearity, memristance and switching mechanism, it can produce a variety of complexity behaviors when building the oscillation circuits [8]. Consequently, the memristor is very suitable for constructing continuous chaotic circuits to generate numerous special behaviors and apply to the engineering applications. So far, many interesting phenomena are reported in the continuous memristor-based chaotic systems, like hyperchaos [9], extremely multistability [10–12], hidden attractors [13–15] or offset-boosting [16]. In addition, some secure communication applications based memristor-based chaotic systems are constantly proposed [17–19]. These studies are basically based on the continuous-time domain, but there are not enough discussions in the discrete-time domain.

As a matter of fact, the structure of discrete nonlinear system is easier for the logic design of digital hardware circuit. And it avoids the disadvantage that parameter sensitivity of the continuous nonlinear system, so the discrete system is more concerned in the practical engineering. Inspired by the mathematical model of the continuous memristor, the discussion of discrete memristor becomes a topical issue in recently [20–26]. He and Peng et al. proposed the integer-order and fractional-order discrete memristor mathematical models, and introduced these discrete models into the sine map [20] and two high-dimensional chaotic maps [21, 22]. Simulations show that these systems can not only enhance the complexity of the original system, but also have abundant dynamic behaviors such as hyperchaos and coexistence attractors. Bao's group presented several hyperchaotic maps based on the discrete memristor [23–25], and verified the numerical results in digital circuits, which laid the foundation for the practical applications of discrete memristor. Kong et al. [26] also reported a new discrete memristive chaotic map, and the offset-boosting and amplitude control are discussed in detail by linear transformation. Although there are more and more studies about the discrete memristive chaotic map, most of them focus on the system modeling and the circuit implementation. The application of the discrete memristive chaotic map has not attracted much attention, especially in its control and synchronization.

Y. Peng

School of Computer Science & School of Cyberspace Science, Xiangtan University, Xiangtan, 411105, P.R.China

S. He, K. Sun

School of Physics and Electronics, Central South University, Changsha, 410083, China

E-mail: hshaobo_123@163.com

Parameter identification is an important step in the control and synchronization of chaotic systems. With the rise of artificial intelligence, it is very popular for utilizing intelligent optimization algorithm to identify the parameters of a chaotic system. It converts the parameter identification into a multi-dimensional optimization problem [27], and the prerequisite is just some time sequences generated by the original chaotic system. Compared with the traditional identification method, it is simple and not sensitive to the considered system. For the Lorenz, Chen, Rössler and other classical continuous chaotic systems, it emerges the differential evolution (DE) algorithm [28,29], particle swarm optimization (PSO) algorithm [30,31], JAYA algorithm [32], artificial bee colony (ABC) algorithm [33], bird swarm algorithm (BSA) [34] to identify the parameters of these systems. In contrast, only a few reports focus on the parameter identification of discrete chaotic maps [35–37]. Due to the stronger sensitivity of the parameters in discrete nonlinear systems, it is a difficult challenge to identify the parameters of discrete chaotic maps, especially for the discrete memristive chaotic maps with coexisting attractors (dynamics guided by initial state) [38]. This motivates us to study an intelligent method to accurately identify the parameters of the discrete memristive chaotic maps with coexisting attractors.

The main contributions of this paper are summarized in the following: 1) a modified differential evolution algorithm called adaptive differential evolution (ADE) algorithm is proposed. 2) The ADE algorithm is used for the parameter identification of two discrete memristive chaotic maps with coexisting attractors. As far as the authors know, this is the first time to study the parameter identification of the discrete memristive chaotic map. 3) In order to solve this complex parameter identification problem, the identification process is divided into two special steps to deal with the global search and the local search, respectively. The remainder of this paper is organized as follows. The preliminaries for parameter identification is introduced in Section 2. Section 3 introduces the ADE algorithm. Numerical simulations are presented in Section 4. Finally, we summarize the conclusion and gives the prospects for the future researches in Section 5.

2 Preliminaries of parameter identification

Preliminary knowledge of parameter identification is introduced in this section. The purpose of parameter identification is to make the parameters of the identification system match the parameters of the original system by minimizing the error of dynamic behavior (i.e. objective function value) between the original system and the identification system. In this paper, two different errors of dynamic behavior are utilized for the identification process, so this section is divided into two subsections: the parameter identification by time sequences and the parameter identification by return maps.

2.1 Parameter identification by time sequences

Parameter identification by time series is the traditional identification method [39]. Its objective function is set to the error of generated time sequences between original system and identification system. The main principle is illustrated in Fig. 1.

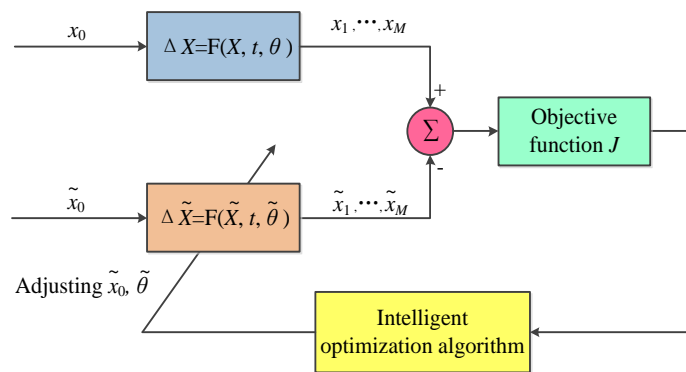


Fig. 1. The main principle of parameter identification by time sequences

Consider the original chaotic map

$$\Delta X = F(X, t, \theta), \quad (1)$$

where $X = (x_1, x_2, \dots, x_M)^T \in R^M$ is the state vector, and M is the number of time sequences for identification ($1 \leq M \leq t$). t is the iteration number, and x_0 is the initial state. $\theta = (\theta_1, \theta_2, \dots, \theta_D)^T \in R^D$ denotes the original system parameter and D is the number of identified system parameter.

The identified system is

$$\Delta \tilde{X} = F(\tilde{X}, t, \tilde{\theta}), \quad (2)$$

where $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_M)^T \in R^M$ denotes the M -dimensional state vector of the identified system. $\tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2, \dots, \tilde{\theta}_D)^T$ is the identified system parameter. The objective function is J , which is calculated by

$$J = \frac{1}{M} \sum_{k=1}^M (X_k - \tilde{X}_k)^2, \quad (3)$$

where X_k and \tilde{X}_k ($k=1,2,\dots,M$) are the state at the k -th time of the original system and the identification system, respectively. Thus the parameter identification by time sequences can be regarded as a multi-dimensional optimization problem which is solved by the intelligent optimization algorithm.

2.2 Parameter identification by return maps

Return maps is new approach for parameter identification problem [40]. Although it also collects a period of time series as the identification samples, its objective function is based on the similarity of chaotic attractors between the original and the identification system. Compared with the traditional parameter identification by time sequences, the return maps approach is less disturbed by the sensitivity of system parameters, but the disadvantage is high time-consuming [41]. Here, the return maps is only used for the local search to deal with the dilemma of local optimum. The main principle of parameter identification by return maps is shown in Fig. 2.

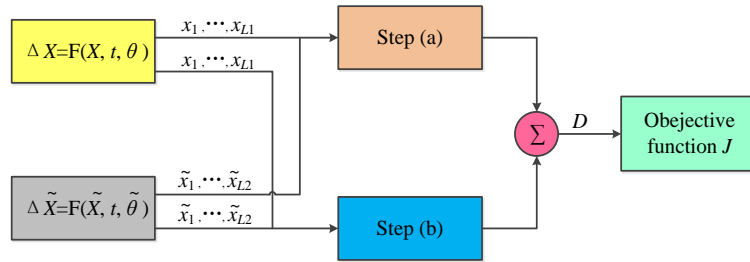


Fig. 2. The main principle of parameter identification by return maps

Consider a original system $\Delta X = F(X, t, \theta)$ and a identified system $\Delta \tilde{X} = F(\tilde{X}, t, \tilde{\theta})$. L_1 and L_2 are the number of time sequences in which the original system and the identification system generate attractor structures, respectively. The meanings of other symbols are consistent with those in the previous subsection. The specific process of Step (a) and Step (b) in the figure is described as follows.

Step (a): For each point in the original system, find their nearest neighbors in the identification system, and calculate the Euclidean distance between them.

Step (b): For each point in the identified system, find their nearest neighbors in the original system, and calculate their Euclidean distance like Step (a).

Add all the Euclidean distances obtained in Step (a) and Step (b) to get the D , and the objective function is calculated by

$$J = D/(L_1 + L_2). \quad (4)$$

The above process is repeated until all the ranges of parameters are searched done. Finally, the identified parameters with objective function of value 0 are the parameters of the original system.

3 Adaptive differential evolution algorithm

Here, the differential evolution (DE) algorithm is employed as the intelligent optimization algorithm for parameter identification. DE algorithm is a famous nature-inspired metaheuristics algorithm for solving multi-dimensional numerical optimization problem [42]. It has four main steps: initialization, mutation, crossover and selection.

3.1 Process of classical DE algorithm

Initialization: The first stage is to initialize the population in the multi-dimensional variable space. Suppose that there are NP biological individuals in the m -dimensional space, where the i -th ($i = 1, 2, \dots, NP$) individual is symbolized by $N_{i,g} = (n_{i,g}^1, n_{i,g}^2, \dots, n_{i,g}^m)$, and g, G is the current number of generation and the maximum number of generation, respectively. The lower search bound and the upper search bound are defined by $N_{min} = (n_{min}^1, n_{min}^2, \dots, n_{min}^m)$ and $N_{max} = (n_{max}^1, n_{max}^2, \dots, n_{max}^m)$, respectively. $N_{i,g}$ is randomly initialized as

$$N_{i,0}^j = n_{min}^j + rand(0, 1)(n_{max}^j - n_{min}^j), \quad (5)$$

where $j = 1, 2, 3, \dots, m$. $rand(0, 1)$ is a random number generated between 0 and 1.

Mutation: In the nature-inspired metaheuristics algorithm, mutation is taken as a sudden change or perturbation with a random element. Each individual generates a new individual by the mutation step, and the new individual is called the mutant vector $V_{i,g}$. The most common mutation definition is

$$V_{i,g} = N_{r_1,g} + F(N_{r_2,g} - N_{r_3,g}), \quad (6)$$

where r_1, r_2 and r_3 are randomly selected from $[1, NP]$. F is the scaling factor, which is a real number between $[0, 1]$.

Crossover: The crossover is employed to generate the trial vectors $U_{i,g} = (u_{i,g}^1, u_{i,g}^2, \dots, u_{i,g}^j)$. In this step, mutant vectors are combined with the original individuals of the current population to form the trial vectors. In general, the binomial recombination is performed on each variable

$$u_{i,g}^j = \begin{cases} v_{i,g}^j, & \text{if } rand(0, 1) \leq CR \text{ or } j = j_{rand}, \\ n_{i,g}^j, & \text{else,} \end{cases} \quad (7)$$

where j_{rand} is randomly selected to ensure that the trial vector could get at least one component from the mutant vector ($j_{rand} \in [1, m]$). CR is called the crossover probability.

Selection: The selection is employed to generate the next generation population. Just like the survival of the fittest in biology, individuals with smaller (or bigger) value are preserved. Obviously, the parameter identification is a minimum optimization problem, so individuals with smaller objective function value are retained, and the selection operation is

$$N_{i,g+1} = \begin{cases} U_{i,g}, & \text{if } f(U_{i,g}) \leq f(N_{i,g}), \\ N_{i,g}, & \text{else,} \end{cases} \quad (8)$$

where $f(\cdot)$ is the objective function.

3.2 Adaptive strategy

The classical DE algorithm doesn't deal with all kinds of optimization problems perfectly. In the face of some high-dimensional complex problems, the classical algorithm would meet the problems of slow convergence rate or low precision [43], so it is still necessary to improve the DE algorithm.

There are two main control parameters of the DE algorithm: the crossover probability CR and the mutation scale factor F . Especially for the value of CR , it is very important to the search process of DE algorithm. Large CR value is more conducive to the global search, but it is easy to premature convergence. On the contrary, small CR is more conducive to local search, but the search rate is relatively slow. According to the aggregation degree of individuals in the search process, an adaptive strategy for CR is proposed as

$$CR = \max\{0.7, 1 - f(Gb)/f(Wb)\}, \quad (9)$$

where Gb and Wb represents the current global optimal vector and the global worst vector, respectively. In addition, according to the situation that the algorithm is easy to fall into premature in the search process, a random mutation factor F is also proposed as

$$F = rand(0, 1)0.1 + 0.45. \quad (10)$$

Based on the two improvements, the new DE algorithm is named the adaptive differential evolution (ADE) algorithm. Finally, the execution process of ADE algorithm is presented in Algorithm 1.

Algorithm 1 Pseudo code of ADE in parameter identification.

```

1: Set the value of  $NP, G, N_{min}, N_{max}$ 
2: Initialize the population according Eq. (5)
3: Calculate the objective function value of each individual, find the  $Gb$  and the  $Wb$ 
4: while  $g < G$  do
5:   Update  $CR$  by Eq. (9) and  $F$  by Eq. (10)
6:   Update the individual's mutant vector and trial vector by Eq. (6) and Eq. (7), respectively
7:   Calculate the individual's objective function value
8:   if  $f(U_{i,g}) \leq f(N_{i,g})$  then
9:      $N_{i,g+1}$  is replaced by  $U_{i,g}$ 
10:  else
11:     $N_{i,g+1}$  is replaced by  $N_{i,g}$ 
12:   $g = g + 1$ 

```

4 Numerical simulations

To verify the effectiveness and advantage of the proposed ADE algorithm, simulations were carried out in two discrete memristive chaotic maps. Five intelligent optimization algorithms, including DE algorithm [43], PSO algorithm [44], JAYA algorithm [45], ABC algorithm [46] and BSA [47] were chosen for the comparison. Simulations were done based on MATLAB 2020a in Intel(R) Core(TM) i7-7700HQ CPU @2.80-GHz with 8 GB RAM.

4.1 Identified systems

The second-order discrete memristive chaotic map is [23]

$$\begin{cases} x_{n+1} = x_n - kx_n \cos(y_n), \\ y_{n+1} = y_n + \varepsilon x_n, \end{cases} \quad (11)$$

where k and ε are two positive system parameters. In general, ε is fixed to $\varepsilon = 1$.

Another second-order discrete memristive chaotic map is [26]

$$\begin{cases} x_{n+1} = a \sin(x_n) \sin(y_n) + (1 - 0.1 |y_n|)x_n, \\ y_{n+1} = 0.4x_n + y_n, \end{cases} \quad (12)$$

where a is the system parameter and $a \neq 1$.

When the parameters of System (11) are set to $k = 2.75, x(0) = 0.5, y(0) = -0.5$ (Lyapunov exponents: 0.1804, 0.1123) and the parameters of System (12) are set to $a = 3.5, x(0) = 1, y(0) = -2$ (Lyapunov exponents: 0.2593, 0.0361), both of the two systems exhibit hyperchaotic behavior, and the phase diagrams of hyperchaotic attractor are shown in Fig. 3. Furthermore, the two discrete memristive chaotic maps have been proved to have the special characteristic of coexisting attractors. So, the dynamic behaviors of the two systems under this set of parameters are very complex. In the following simulations, these parameters $k = 2.75, x_0 = 0.5, y_0 = -0.5$ and $a = 3.5, x_0 = 1, y_0 = -2$ were selected as the original system parameters.

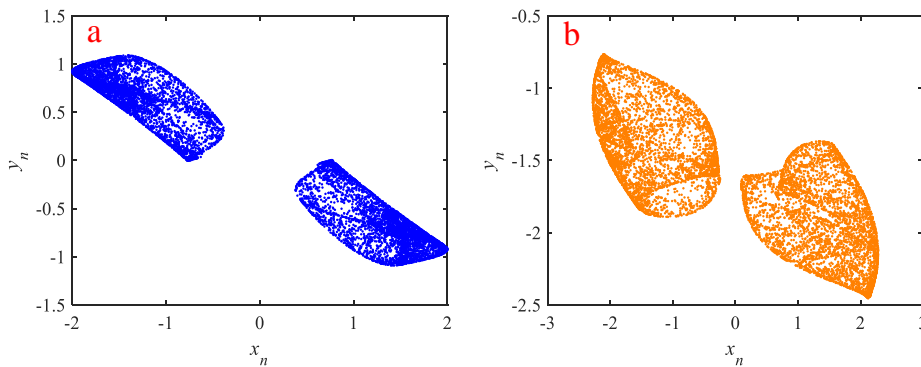


Fig. 3. Hyperchaotic attractor for (a) system (11) with parameters $k = 2.75, x(0) = 0.5, y(0) = -0.5$ and (b) system (12) with parameters $a = 3.5, x(0) = 1, y(0) = -2$

4.2 Parameter identification by different algorithms

In this subsection, the parameter identification by time sequences was carried out under six intelligent optimization algorithms. The maximum number generation of each algorithms was set to $G = 100$, and other settings are presented in Table 1. The number of time sequences was $M = 3$. To eliminate the stochastic nature of the algorithms, we performed 100 consecutive algorithm runs. The search ranges of the original system parameters were set as: $0 \leq k \leq 4$, $-4 \leq x_0 \leq -4$, $-4 \leq y_0 \leq 4$ for system (11) and $0 \leq a \leq 4$, $-4 \leq x_0 \leq -4$, $-4 \leq y_0 \leq 4$ for system (12).

Table 1. Parameter settings of the intelligent optimization algorithms

Algorithm	Parameters
DE [43]	$NP = 40, F = 0.6, CR = 0.6$
PSO [44]	$NP = 40, c_1 = c_2 = 2, \omega_{max} = 9.0, \omega_{min} = 0.4$
JAYA [45]	$NP = 40$
ABC [46]	$NP = 40, limit = 100$
BSA [47]	$NP = 40, C = S = 1.5, a_1 = a_2 = 1, FQ = 3, P \in [0.8, 1], FL \in [0.5, 0.9]$

The simulation results for parameter identification by time sequences under different algorithms are listed in Table 2. The statistical results of the best J value, average J value and accurate identification rate are given in this table. Generally, the smaller the value of objective function J is, the higher the identification accuracy is. When the error between each identification parameters and the corresponding original parameters is less than $1e-04$, it is considered as the accurate identification. From Table 2, it is shown that DE, PSO, BSA and ADE algorithms can accurately identify the parameters of the original system. But the JAYA algorithm and ABC algorithm are invalid for the two discrete memristive chaotic maps. According to the accurate identification rate, the ADE algorithm has the best precision, and the accuracy of the modified algorithm is much higher than that of the classical DE algorithm. PSO algorithm and BSA algorithm are ranked in the second and the third, respectively. In addition, it should be noticed that the identification difficulty of system (11) is higher than that of system (12).

Table 2. Parameter identification by different algorithms

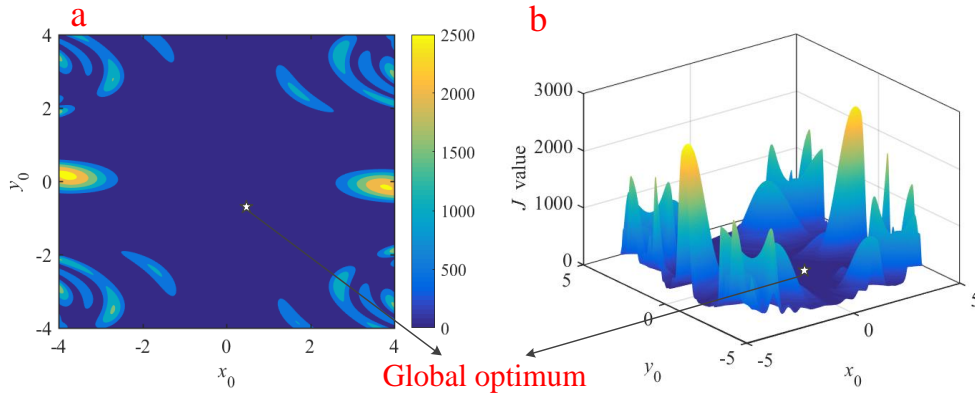
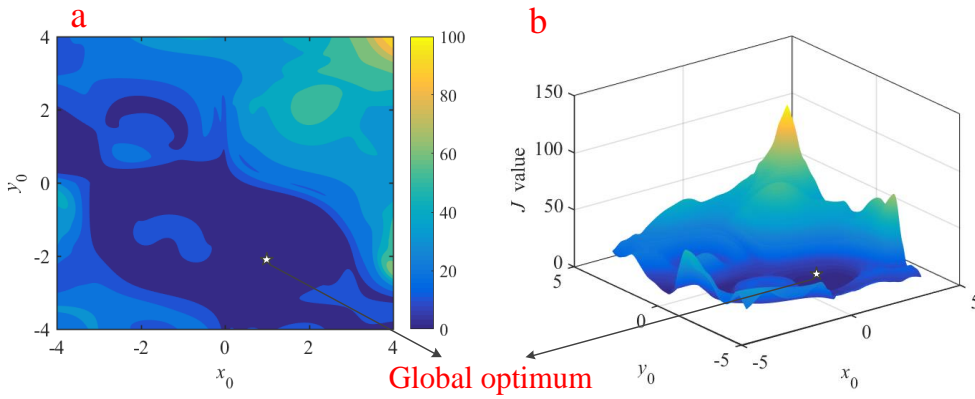
Chaotic map	Algorithm	Best J value	Average J value	Accurate identification rate
System (11)	DE	6.7144e-10	1.2865e-06	3%
	PSO	1.2290e-14	7.4035e-11	45%
	JAYA	4.4898e-03	1.3961e-01	0
	ABC	1.8601e-05	8.9198e-04	0
	BSA	1.6935e-14	1.5328e-03	26%
	ADE	1.1950e-21	4.1163e-15	66%
System (12)	DE	2.5904e-10	2.2351e-06	1%
	PSO	2.5824e-14	6.0965e-11	28%
	JAYA	2.9983e-03	1.0712e-01	0
	ABC	2.0735e-06	3.5651e-04	0
	BSA	6.7133e-14	1.4658e-06	10%
	ADE	1.5215e-20	1.5768e-14	49%

Although four algorithms have probability to accurately identify the parameters of the original system, they inevitably fall into the local optimum (i.e. the final identified parameters can not converge to the original parameters). To check the specific identification results, the final identified parameters of the ADE, PSO and BSA are listed in Table 3. It shows that these algorithms are very accurate for the identification of system parameters, but there are many local optimal solutions for the initial state identification. The PSO algorithm and BSA algorithm have more than two local optimal solutions, while the ADE algorithm has only one local optimal solution. Therefore, the ADE algorithm has higher accuracy and better stability.

When the system parameters is fixed to $k = 2.75$, $a = 3.5$ and the initial states are set as variable, the J value of parameter identification by time sequences for system (11) and (12) are plotted in Fig. 4 and 5, respectively. In the figures, dark blue indicates that the value of the objective function is low (i.e. the smaller error). The figures show that there is a large area of local optimal region near the global optimum, and the difference of J value between local optimal region and global optimum is only about $1e-02$ to $1e-05$. Thus, due to the special property of coexisting attractors in the two maps, the change of initial state has big influence on the dynamic behavior, resulting in the situation of multiple local optimal solutions. To solve this problem, the return maps approach is introduced in the following.

Table 3. Identified parameters by ADE, PSO and BSA algorithms

Chaotic map	Algorithm	Identified parameters	Probability
System (11)	ADE	$\tilde{k} = 2.7500, \tilde{x}_0 = 0.5000, \tilde{y}_0 = -0.5000$	66%
		$\tilde{k} = 2.7500, \tilde{x}_0 = 0.8344, \tilde{y}_0 = -0.8344$	34%
	PSO	$\tilde{k} = 2.7500, \tilde{x}_0 = 0.5000, \tilde{y}_0 = -0.5000$	45%
		$\tilde{k} = 2.7500, \tilde{x}_0 = 0.8344, \tilde{y}_0 = -0.8344$	37%
	BSA	$\tilde{k} = 2.7500, \tilde{x}_0 = -1.3909, \tilde{y}_0 = 1.3909$	18%
		$\tilde{k} = 2.7500, \tilde{x}_0 = 0.5000, \tilde{y}_0 = -0.5000$	26%
System (12)	ADE	$\tilde{a} = 3.5000, \tilde{x}_0 = 1.0000, \tilde{y}_0 = -2.0000$	49%
		$\tilde{a} = 3.5000, \tilde{x}_0 = 1.2534, \tilde{y}_0 = -2.1014$	51%
	PSO	$\tilde{a} = 3.5000, \tilde{x}_0 = 1.0000, \tilde{y}_0 = -2.0000$	28%
		$\tilde{a} = 3.5000, \tilde{x}_0 = 1.2534, \tilde{y}_0 = -2.1014$	35%
	BSA	$\tilde{a} = 3.4999, \tilde{x}_0 = -2.7196, \tilde{y}_0 = -0.5121$	37%
		$\tilde{a} = 3.5000, \tilde{x}_0 = 1.0000, \tilde{y}_0 = -2.0000$	10%
	$\tilde{a} = 3.5000, \tilde{x}_0 = 1.2534, \tilde{y}_0 = -2.1014$	41%	
	$\tilde{a} = 3.4999, \tilde{x}_0 = -2.7196, \tilde{y}_0 = -0.5121$	38%	
	Others		11%

**Fig. 4.** The objective function J value of parameter identification by time sequences for system (11) (a) two-dimensional contour figure (b) three-dimensional contour figure**Fig. 5.** The objective function J value of parameter identification by time sequences for system (12) (a) two-dimensional contour figure (b) three-dimensional contour figure

4.3 Hybrid method for parameter identification

Because there is the local optimal solution problem when using time sequences for parameter identification, the return maps method is employed for the local search. Due to the better performance of ADE algorithm, other

algorithms are not considered in this subsection. Thus, a hybrid method is proposed (ADE-Return maps), which consists of two steps: the ADE algorithm of parameter identification by time sequences is utilized for global search, and the parameter identification by return maps is utilized for local search. Its specific implementation process is presented in Algorithm 2. The number of time sequences for constructing attractor shapes was set to $M = 300$.

Algorithm 2 Pseudo code of ADE-Return maps in parameter identification.

- 1: Execute Algorithm 1
 - 2: Find the global optimal vector Gb
 - 3: $low = \text{roundn}(Gb, 2) - 0.5$, $upp = \text{roundn}(Gb, 2) + 0.5$
 - 4: $C = low : 0.01 : upp$
 - 5: **for** $c = \text{length}(C)$ **do**
 - 6: Calculate the objective function value $J = f(c)$
 - 7: Find the smallest objective function value $\min(J)$
 - 8: Find the new global optimal vector Gb
-

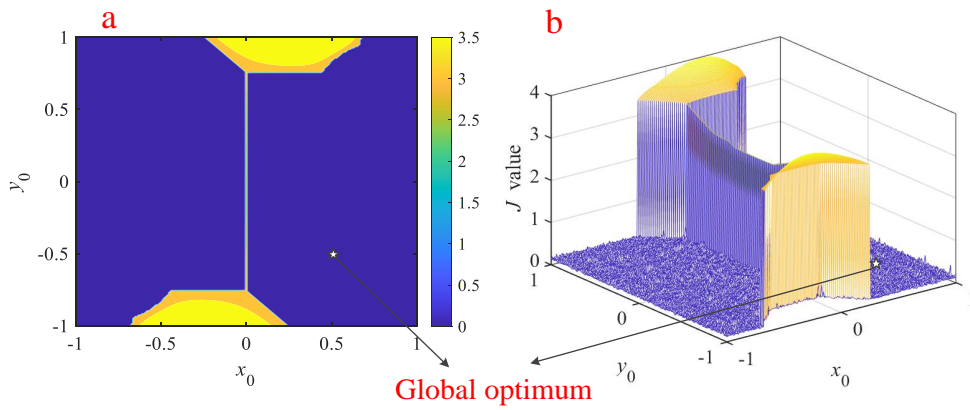


Fig. 6. The objective function J value of parameter identification by return maps for system (11) (a) two-dimensional contour figure (b) three-dimensional contour figure

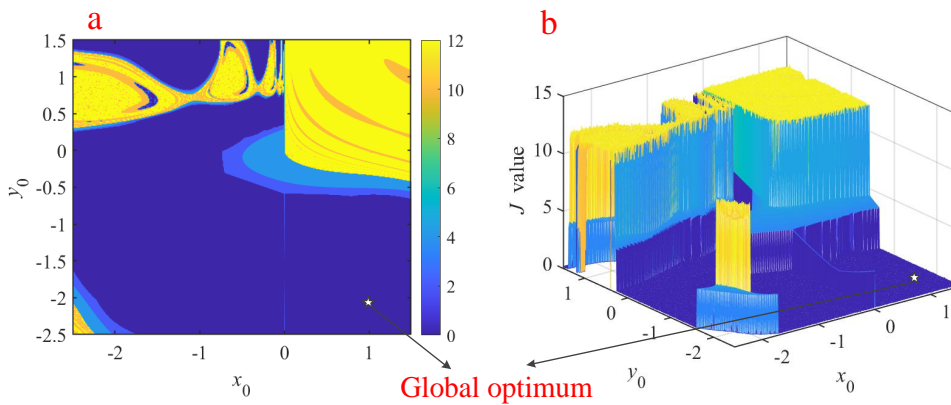


Fig. 7. The objective function J value of parameter identification by return maps for system (12) (a) two-dimensional contour figure (b) three-dimensional contour figure

Similarly, when the system parameters are fixed and the initial states are set to the variable, the J value of parameter identification by return maps for system (11) and (12) are plotted in Fig. 6 and 7, respectively. Obviously, the change of J value in return maps method is more drastic than that of the parameter identification by time sequences (it's similar to the difference between the high-rise buildings in the city and the rolling hills). It is because the change of the initial states leads to the sudden change of the attractor structure, which is far

greater than the error between time sequences. And the difference of J value between local optimal region and global optimum is about 0.1, this is much larger than that of the parameter identification by time sequences.

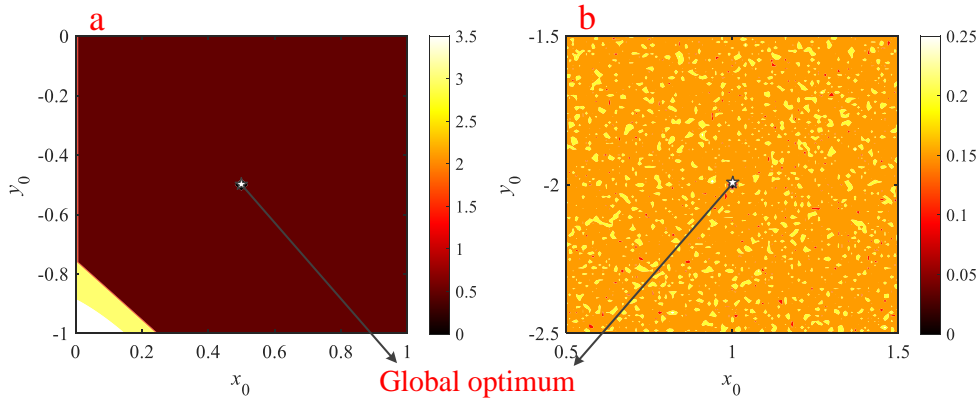


Fig. 8. Parameter identification by ADE-Return maps for (a) system (11) (b) system (12)

As shown in Table 3, there is always only one local optimal solution by the ADE algorithm, so the local search by return maps can be carried out according to the local optimal solution (here the search range is set to the local optimal solution ± 0.5). Hence, such a mechanism ensures that the ADE-Return maps method can find the global optimal solution, as shown in the white star in Fig 8.

4.4 Noise interference

Numerical simulation of parameter identification by ADE-Return maps under noise interference is investigated in this subsection. The additive white Gaussian noise (AWGN) is considered in the time sequences and the attractor structure obtained by the original system.

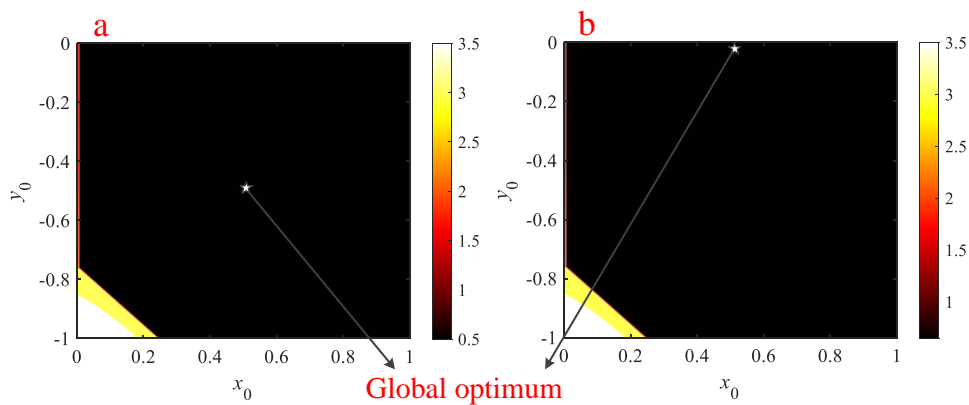


Fig. 9. Parameter identification for system (11) with noise (a) SNR 20 dB (b) SNR 15 dB

The numerical simulation results for system (11) are presented in Fig. 9. It is illustrated that the ADE-Return maps successfully finds the global optimum (namely $x_0 = 0.5, y_0 = -0.5$) with the signal-to-noise ratio (SNR) 20dB. However, it cannot find the correct global optimum with SNR 15dB. The results for system (12) are illustrated in Fig. 10. It also shows that the ADE-Return maps can identify the global optimum (namely $x_0 = 1, y_0 = -2$) until the SNR reaches 15dB. The experiments show that the proposed approach has good anti-noise performance. For the two discrete memristive chaotic maps, it can resist noise larger than SNR 20dB.

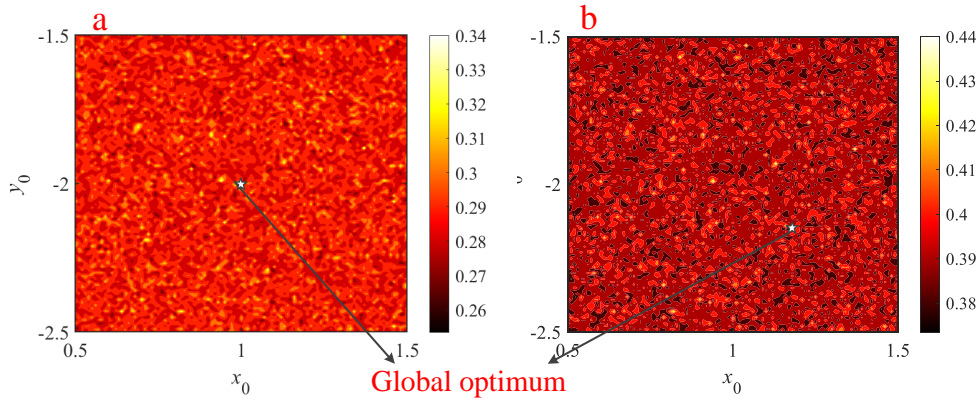


Fig. 10. Parameter identification for system (12) with noise (a) SNR 20 dB (b) SNR 15 dB

5 Conclusion

In this paper, an improved intelligent optimization algorithm called ADE algorithm is proposed for the parameter identification. The target system is two discrete memristive chaotic maps with complex hyperchaotic behaviors and coexisting attractors. In order to solve the problem of falling into the local optimal solutions, the ADE algorithm based on the two objective function of time sequences error and return maps is employed. Numerical simulation results lead to the following conclusions.

(a) For the parameter identification by time sequences, the proposed ADE algorithm outperforms the other five intelligent optimization algorithms, due to the highest identification accuracy and only one local optimal solution.

(b) By utilizing the return maps as the objective function, the ADE algorithm successfully jumps out of the local optimal solution and find the global optimum.

(c) After considering the interference of AWGN, the proposed approach still finds the global optimum when SNR reaches 15dB.

In the numerical simulations, the factors in real-world applications are studied, including unknown system parameters, unknown initial values and noise. These simulations were therefore theoretically constitute a possibility for the synchronization control of the discrete memristive chaotic maps. What is more important is that the introduced approach may be a useful tool for the future applications of discrete memristor. Our next work is to try to apply this method in the secure communication.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant Nos. 62071496, 62061008 and 61901530), the China Postdoctoral Science Foundation (No.2019M652791) and the Hunan Provincial Education Department, China (No. 20C1787).

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

1. Leon Chua. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, 18:507–519, 1971.
2. Dmitri Strukov, Gregory Snider, Duncan Stewart, and Stanley Williams. The missing memristor found. *Nature*, 453:80–83, 2008.
3. Bo Zhao, Ming Xiao, and Norman Zhou. Synaptic learning behavior of a tio2 nanowire memristor. *Nanotechnology*, 30:425202, 2019.
4. Jun Ma and Jun Tang. A review for dynamics in neuron and neuronal network. *Nonlinear Dynamics*, 89:1569–1578, 2017.
5. Chunhua Wang, Lin Xiong, Jingru Sun, and Wei Yao. Memristor-based neural networks with weight simultaneous perturbation training. *Nonlinear Dynamics*, 95:2893–2906, 2019.
6. Jodh Singh Pannu, Sunny Raj, Steven Lawrence Fernandes, Dwaipayana Chakraborty, Sarah Rafiq, Nathaniel Cady, and Sumit Kumar Jha. Design and fabrication of flow-based edge detection memristor crossbar circuits. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67:19589441, 2020.

7. Qiang Lai, Zhiqiang Wan, Paul Didier Kamdem Kuate, and Hilaire Fotsin. Coexisting attractors, circuit implementation and synchronization control of a new chaotic system evolved from the simplest memristor chaotic circuit. *Communications in Nonlinear Science and Numerical Simulation*, 89:105341, 2020.
8. Makoto Itoh and Leon Chua. Dynamics of memristor circuits. *International Journal of Bifurcation and Chaos*, 24:1430015, 2014.
9. M. Emin Sahin, A. Samil Demirkol, Hasan Guler, and Serdar E. Hamamci. Design of a hyperchaotic memristive circuit based on wien bridge oscillator. *Computers & Electrical Engineering*, 88:106826, 2020.
10. Han Bao, Ning Wang, Bocheng Bao, Mo Chen, Peipei Jin, and Guangyi Wang. Initial condition-dependent dynamics and transient period in memristor-based hypogenetic jerk system with four line equilibria. *Communications in Nonlinear Science and Numerical Simulation*, 57:264–275, 2018.
11. Yunzhen Zhang, Zhong Liu, Huagan Wu, Shengyao Chen, and Bocheng Bao. Two-memristor-based chaotic system and its extreme multistability reconstitution via dimensionality reduction analysis. *Chaos Solitons & Fractals*, 127:354–363, 2019.
12. Hui Chang, Yuxia Li, Guanrong Chen, and Fang Yuan. Extreme multistability and complex dynamics of a memristor-based chaotic system. *International Journal of Bifurcation and Chaos*, 30:2030019, 2020.
13. Karthikeyan Rajagopal, Anitha Karthikeyan, and Ashokkumar Srinivasan. Dynamical analysis and fpga implementation of a chaotic oscillator with fractional-order memristor components. *Nonlinear Dynamics*, 91:1491–1512, 2018.
14. Vaibhav Varshney, S. Sabarathinam, Awadhesh Prasad, and K. Thamilmaran. Infinite number of hidden attractors in memristor-based autonomous duffing oscillator. *International Journal of Bifurcation and Chaos*, 28:1850013, 2018.
15. Sen Zhang, Jiahao Zheng, Xiaoping Wang, and Zhigang Zeng. Multi-scroll hidden attractor in memristive hr neuron model under electromagnetic radiation and its applications. *Chaos*, 31:011101, 2021.
16. Jiacheng Gu, Chunbiao Li, Yudi Chen, Herbert H. C. Iu, and Tengfei Lei. A conditional symmetric memristive system with infinitely many chaotic attractors. *IEEE Access*, 8:12394–12401, 2020.
17. Yang Yang, Lidan Wang, Shukai Duan, and Li Luo. Dynamical analysis and image encryption application of a novel memristive hyperchaotic system. *Optics and Laser Technology*, 133:106553, 2021.
18. Viet-Thanh Pham, Christos Volos, and Luigi Fortuna. Memristor-based systems: nonlinearity, dynamics and applications. *European Physical Journal-Special Topics*, 228:1902–1906, 2019.
19. Jian Liu and Rui Xu. Adaptive synchronisation of memristor-based neural networks with leakage delays and applications in chaotic masking secure communication. *International Journal of Systems Science*, 49:1300–1315, 2018.
20. Shaobo He, Kehui Sun, Yuexi Peng, and Lingyu Wang. Modeling of discrete fracmemristor and its application. *AIP Advances*, 10:015332, 2020.
21. Yuexi Peng, Shaobo He, and Kehui Sun. A higher dimensional chaotic map with discrete memristor. *AEU-International Journal of Electronics and Communications*, 129:153539, 2021.
22. Yuexi Peng, Shaobo He, and Kehui Sun. Chaos in the discrete memristor-based system with fractional-order difference. *Results in Physics*, 24:104106, 2021.
23. Bocheng Bao, Houzhen Li, Huagan Wu, Xi Zhang, and Mo Chen. Hyperchaos in a second-order discrete memristor-based map model. *Electronics Letters*, 56:769–770, 2020.
24. Houzhen Li, Zhongyun Hua, Han Bao, Lei Zhu, Mo Chen, and Bocheng Bao. Two-dimensional memristive hyperchaotic maps and application in secure communication. *IEEE Transactions on Industrial Electronics*, page Early Access., 2020.
25. Bocheng Bao, Kang Rong, Houzhen Li, Kexin Li, Zhongyun Hua, and Xi Zhang. Memristor-coupled logistic hyperchaotic map. *IEEE Transactions on Circuits and Systems II: Express Briefs*, page Early Access., 2021.
26. Sixiao Kong, Chunbiao Li, Shaobo He, Serdar Cicek, and Qiang Lai. A memristive map with coexisting chaos and hyperchaos. *Chinese Physics B*, page Early Access., 2021.
27. D. A. Yousri, Amr M. AbdelAty, Lobna A. Said, A. S. Elwakil, Brent Maundy, and Ahmed G. Radwan. Parameter identification of fractional-order chaotic systems using different meta-heuristic optimization algorithms. *Nonlinear Dynamics*, 95:2491–2542, 2019.
28. Wei Du, Qingying Miao, Le Tong, and Yang Tang. Identification of fractional-order systems with unknown initial values and structure. *Physics Letters A*, 381:1943–1949, 2017.
29. Wenjuan Gu, Yongguang Yu, and Wei Hu. Parameter estimation of unknown fractional-order memristor-based chaotic systems by a hybrid artificial bee colony algorithm combined with differential evolution. *Nonlinear Dynamics*, 84:779–795, 2016.
30. Juan A. Lazzus, Pedro Vega-Jorquera, Carlos H. Lopez-Caraballo, Luis Palma-Chilla, and Ignacio Salfate. Parameter estimation of a generalized lotka-volterra system using a modified pso algorithm. *Applied Soft Computing*, 96:106606, 2020.
31. Yuexi Peng, Kehui Sun, Shaobo He, and Xi Yang. Parameter estimation of a complex chaotic system with unknown initial values. *European Physical Journal Plus*, 133:305, 2018.
32. Feixin Chen, Zhenghao Ding, Zhongrong Lu, and Xiangkun Zeng. Parameters identification for chaotic systems based on a modified jaya algorithm. *Nonlinear Dynamics*, 94:2307–2326, 2018.
33. Sangeeta Gupta, Varun Upadhyaya, Ayush Singh, Pragya Varshney, and Smriti Srivastava. Modeling of fractional order chaotic systems using artificial bee colony optimization and ant colony optimization. *Journal of Intelligent & Fuzzy Systems*, 35:5337–5344, 2018.
34. Pei Zhang, Renyu Yang, Renhuan Yang, Gong Ren, and Xiuzeng Yang et al. Parameter estimation for fractional-order chaotic systems by improved bird swarm optimization algorithm. *International Journal of Modern Physics C*, 30:1950086, 2019.
35. Adel Ouannas, Amina-Aicha Khennaoui, Giuseppe Grassi, and Samir Bendoukhad. On chaos in the fractional-order grassi–miller map and its control. *Journal of Computational and Applied Mathematics*, 358:293–305, 2019.
36. Yuexi Peng, Kehui Sun, Shaobo He, and Dong Peng. Parameter identification of fractional-order discrete chaotic systems. *Entropy*, 21:27, 2019.
37. Yuexi Peng, Kehui Sun, and Shaobo He. Dynamics analysis of chaotic maps: From perspective on parameter estimation by meta-heuristic algorithm. *Chinese Physics B*, 29:030502, 2020.
38. Sajad Jafari, Atefeh Ahmadi, Shirin Panahi, and Karthikeyan Rajagopal. Extreme multi-stability: When imperfection changes quality. *Chaos Solitons & Fractals*, 108:182–186, 2018.
39. Giuseppe Quaranta, Walter Lacarbonara, and Sami F. Masri. A review on computational intelligence for identification of nonlinear dynamical systems. *Nonlinear Dynamics*, 99:1709–1761, 2020.
40. Sajad Jafari, Julien C. Sprott, Viet-Thanh Pham, S. Mohammad Reza Hashemi Golpayegani, and Amir Homayoun Jafari. A new cost function for parameter estimation of chaotic systems using return maps as fingerprints. *International Journal of Bifurcation and Chaos*, 24:1450134, 2014.
41. Yuexi Peng, Kehui Sun, and Shaobo He. An improved return maps method for parameter estimation of chaotic systems. *International Journal of Bifurcation and Chaos*, 30:2050058, 2020.
42. Rainer Martin Storn and Kenneth Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *ICSI Technical Report TR-95-012.*, 1995.
43. Swagatam Das and Ponnuthurai Nagarathnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15:4–31, 2011.

44. Dongshu Wang, Dapei Tan, and Lei Liu. Particle swarm optimization algorithm: an overview. *Soft Computing*, 22:387–408, 2018.
45. R. Venkata Rao. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 7:19–34, 2016.
46. Dervis Karaboga and Bahriye Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214:108–132, 2009.
47. Xianbing Meng, X.z. Gao, Lihua Lu, Yu Liu, and Hengzhen Zhang. A new bio-inspired optimisation algorithm: Bird swarm algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, 28:673–687, 2016.