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

Yuexi Peng · Shaobo He · Kehui Sun

Received: 19 July 2021 / Accepted: 10 October 2021 / Published online: 25 October 2021
© The Author(s), under exclusive licence to Springer Nature B.V. 2021

Abstract Since the concept of discrete memristor was proposed, more and more scholars began to study this topic. At present, most of works on the discrete memristor are devoted to the mathematical modeling and digital 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 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 multi-stability [10, 11], hidden attractors [12, 13] or offset-boosting [14, 15]. In addition, some secure communication applications based on memristor-based chaotic systems are constantly proposed [16, 17]. 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 of param-
eter sensitivity in continuous nonlinear system, so the discrete system is more concerned in the practical engineering. Inspired by the mathematical model of continuous memristor, the discussion of discrete memristor becomes a topical issue in recently [18–26]. 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 [18] and two high-dimensional chaotic maps [19,20]. 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 [21–25] 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.

Chaos synchronization [27] has great significance in the control of chaotic systems, and it plays an important role in chaotic secure communication. In many real-world applications, the parameters cannot be directly obtained, so parameter identification becomes the key step for the successful synchronization control 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 typical multi-dimensional optimization problem [28], 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 [29], particle swarm optimization (PSO) algorithm [30], JAYA algorithm [31], artificial bee colony (ABC) algorithm [32], bird swarm algorithm (BSA) [33] to identify the parameters of these systems. In contrast, only a few reports focus on the parameter identification of discrete chaotic maps [34–36]. Due to the stronger sensitivity of the parameters in discrete nonlinear systems, it is a challenge to identify the parameters of discrete chaotic maps, especially for the discrete memristive chaotic maps with coexisting attractors (dynamics guided by the initial state) [37]. This motivates us to develop 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: (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. 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 are introduced in Sect. 2. Section 3 introduces the ADE algorithm. Numerical simulations are presented in Sect. 4. Finally, we summarize the conclusion and gives the prospects for the future researches in Sect. 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 [38]. Its objective function is set to the error of generated time sequences between orig-
Parameter identification for discrete memristive

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

As presented in Fig. 1, considering there is an original chaotic map

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

where $$X = (x_1, x_2, ..., x_M)^T \in R^M$$ 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. $$\theta = (\theta_1, \theta_2, \ldots, \theta_D)^T \in R^D$$ denotes the original system parameter and $$D$$ is the number of identified system parameter.

The identification system is

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

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

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

where $$X_k$$ and $$\tilde{X}_k$$ ($$k=1,2,...,M$$) 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.

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

2.2 Parameter identification by return maps

Return maps is new approach for parameter identification problem [39]. Although it also collects a period of time sequences as the identification sample, 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 [40]. 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.

Considering an original system $$\Delta X = F(X, t, \theta)$$ and an identification 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 meaning of other symbols is consistent with those in Sect. 2.1. 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 identification 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 sum of distance $$D$$, and the objective function is calculated by

$$J = D / (L_1 + L_2).$$

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 optimization problem [41]. 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, ..., NP) individual is symbolized by \( Ni,g = (n_{i,g}^1, n_{i,g}^2, ..., n_{i,g}^m) \), and g, G is the current number and the maximum number of generation, respectively. The lower search bound and the upper search bound are defined by \( N_{min} = (n_{min}^1, n_{min}^2, ..., n_{min}^m) \) and \( N_{max} = (n_{max}^1, n_{max}^2, ..., n_{max}^m) \), respectively. \( N_i,g \) is randomly initialized as

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

where \( j = 1, 2, 3, ..., m \). 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 the 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

\[
V_{i,g} = N_{r1,g} + F(N_{r2,g} - N_{r3,g}), \tag{6}
\]

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 vector \( U_{i,g} = (u_{i,g}^1, u_{i,g}^2, ..., u_{i,g}^m) \). 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

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

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 retained, and the selection operation is

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

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

3.2 Adaptive strategy

According to the “No free lunch” theorems [42], the classical DE algorithm cannot handle all kinds of optimization problems well. 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 get 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

\[
CR = \max\{0.7, 1 - f(Gb)/f(Wb)\}, \tag{9}
\]

where \( Gb \) and \( Wb \) represent 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

\[
F = F_{min} + (F_{max} - F_{min}) \cdot rand(0, 1),
\]

where \( F_{min} \) and \( F_{max} \) are the minimal and maximal mutation factor, respectively.
\[
F = \text{rand}(0, 1) \times 0.1 + 0.45. \tag{10}
\]

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 \( NP, G, N_{\text{min}}, N_{\text{max}} \).
2. Initialize the population according Eq. (5).
3. Calculate the objective function value of each individual, find the \( Gb \) and the \( Wh \).
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 [21]

\[
\begin{align*}
    x_{n+1} &= x_n - k x_n \cos(y_n), \\
    y_{n+1} &= y_n + \varepsilon x_n,
\end{align*}
\tag{11}
\]

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

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

\[
\begin{align*}
    x_{n+1} &= a \sin(x_n) \sin(y_n) + (1 - 0.1 | y_n |) x_n, \\
    y_{n+1} &= 0.4 x_n + y_n,
\end{align*}
\tag{12}
\]

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.

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 algorithm was set to \( G = 100 \), and other settings are presented in Table 1. The sample number of time sequences for parameter identification 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: \( 0 \leq k \leq 4, -4 \leq x_0, y_0 \leq 4 \) for system (11) and \( 0 \leq a \leq 4, -4 \leq x_0, y_0 \leq 4 \) for system (12).

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
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 $a = 3.5$, $x(0) = 1$, $y(0) = -2$

Table 1 Parameter settings of the intelligent optimization algorithms

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

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%**                      |

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
Parameter identification for discrete memristive 1269

Table 3 Identified parameters by ADE, PSO and BSA algorithms

| Chaotic map | Algorithm | Identified parameters | Probability |
|-------------|-----------|-----------------------|-------------|
| System (11) | ADE       | $\hat{k} = 2.7500$, $\hat{x}_0 = 0.5000$, $\hat{y}_0 = -0.5000$ | 66%         |
|            |           | $\hat{k} = 2.7500$, $\hat{x}_0 = 0.8344$, $\hat{y}_0 = -0.8344$ | 34%         |
|            | PSO       | $\hat{k} = 2.7500$, $\hat{x}_0 = 0.5000$, $\hat{y}_0 = -0.5000$ | 45%         |
|            |           | $\hat{k} = 2.7500$, $\hat{x}_0 = 0.8344$, $\hat{y}_0 = -0.8344$ | 37%         |
|            | BSA       | $\hat{k} = 2.7500$, $\hat{x}_0 = -1.3909$, $\hat{y}_0 = 1.3909$ | 18%         |
|            |           | $\hat{k} = 2.7500$, $\hat{x}_0 = 0.5000$, $\hat{y}_0 = -0.5000$ | 26%         |
|            |           | $\hat{k} = 2.7500$, $\hat{x}_0 = 0.8344$, $\hat{y}_0 = -0.8344$ | 34%         |
|            |           | $\hat{k} = 2.7500$, $\hat{x}_0 = -1.3909$, $\hat{y}_0 = 1.3909$ | 16%         |
|            |           | Others | 24%         |
| 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%         |

third, respectively. In addition, it should be noticed that the identification difficulty of system (12) is higher than that of system (11).

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 cannot converge to the original parameters). To check the detailed identification results, the final identified parameters of the ADE, PSO and BSA are given 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 are 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) is plotted in Figs. 4 and 5, respectively. In these figures, dark blue indicates that the value of the objective function is low (i.e., the smaller error). The figures illustrate 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.

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 = roundn($Gb$, 2) - 0.5, upp = roundn($Gb$, 2) + 0.5
4: C = low: 0.01 : upp
5: for $c$ = 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. 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
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) is plotted in Figs. 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 is similar to the difference between the high-rise buildings in the city and the rolling hills). The reason is that 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.

Furthermore, the difference of $J$ value between local optimal region and global optimum is about 0.1; this is also much larger than that of the parameter identification by time sequences. Therefore, the global optimal solution can be easily identified from other solutions in this region.

As listed 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 $\pm 0.5$, which is given...
in Algorithm 2). Hence, such the 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. Noise is a troublesome factor that cannot be ignored in practical engineering, so the additive white Gaussian noise (AWGN) was considered in the time sequences and the attractor structure obtained by the original system.

The numerical simulation results for system (11) are presented in Fig. 9. It is illustrated that the ADE-Return maps successfully find the global optimum (namely $x_0 = 0.5, y_0 = -0.5$) with the signal-to-noise ratio (SNR) 20 dB. However, it cannot find the correct global optimum with SNR 15 dB. 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 15 dB. 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 20 dB.
Parameter identification for discrete memristive

Fig. 10 Parameter identification by ADE-Return maps for system (12) with noise a SNR 20 dB b SNR 15 dB

In contrast, if the objective function of ADE-Return maps method is changed into the error of time sequences (called ADE-Time sequences), the identification results under AWGN interference are given in Fig. 11. The figure shows that the algorithm cannot find the global optimal solution for systems (11) and (12) when the SNR only reaches 45 and 40 dB, respectively. This is also a reason why the RM method is chosen for the local search.

5 Conclusion

In this paper, an improved intelligent optimization algorithm called ADE algorithm is proposed for the parameter identification. The target system is selected as 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 until 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 The authors would like to thank the anonymous reviewers for their constructive comments and insightful suggestions.

Funding Information This work is supported by the National Natural Science Foundation of China (Grant Nos. 62071496, 62061008 and 61901530), the Natural Science Foundation of Hunan Province (No.2021JJ40545) and the Hunan Provincial Education Department (No. 20C1787).

Data availability statement The datasets (MATLAB codes) generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declaration

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Chua, L.: Memristor-the missing circuit element. IEEE Trans. Circuit Theory 18, 507–519 (1971)
2. Strukov, D., Snider, G., Stewart, D., Williams, S.: The missing memristor found. Nature 453, 80–83 (2008)
3. Zhao, B., Xiao, M., Zhou, N.: Synaptic learning behavior of a tio2 nanowire memristor. Nanotechnology 30, 425202 (2019)
4. Ma, J., Tang, J.: A review for dynamics in neuron and neuronal network. Nonlinear Dyn. 89, 1569–1578 (2017)
5. Wang, C., Xiong, L., Sun, J., Yao, W.: Memristor-based neural networks with weight simultaneous perturbation training. Nonlinear Dyn. 95, 2893–2906 (2019)
6. Pannu, J.S., Raj, S., Fernandes, S.L., Chakraborty, D., Rafaq, S., Cady, N., Jha, S.K.: Design and fabrication of flow-based edge detection memristor crossbar circuits. IEEE Trans. Circuits Syst. II: Express Briefs 67, 19589441 (2020)
7. Lai, Q., Wan, Z., Kuate, P.D.K., Fotsin, H.: Coexisting attractors, circuit implementation and synchronization control of a new chaotic system evolved from the simplest memristor chaotic circuit. Commun. Nonlinear Sci. Numer. Simul. 89, 105341 (2020)
8. Itoh, M., Chua, L.: Dynamics of memristor circuits. Int. J. Bifurcat. Chaos 24, 1430015 (2014)
9. Sahin, M.E., Demirkol, A.S., Guler, H., Hamamci, S.E.: Design of a hyperchaotic memristive circuit based on wien bridge oscillator. Comput. Electr. Eng. 88, 106826 (2020)
10. Bao, H., Wang, N., Bao, B., Chen, M., Jin, P., Wang, G.: Initial condition-dependent dynamics and transient period in memristor-based hypogenetic jerk system with four line equilibria. Commun. Nonlinear Sci. Numer. Simul. 57, 264–275 (2018)
11. Chang, H., Li, Y., Chen, G., Yuan, F.: Extreme multistability and complex dynamics of a memristor-based chaotic system. Int. J. Bifurcat. Chaos 30, 2030019 (2020)
12. Rajagopal, K., Karthikeyan, A., Srinivasan, A.: Dynamical analysis and FPGA implementation of a chaotic oscillator with fractional-order memristor components. Nonlinear Dyn. 91, 1491–1512 (2018)
13. Zhang, S., Zheng, J., Wang, X., Zeng, Z.: Multi-scroll hidden attractor in memristive hr neuron model under electromagnetic radiation and its applications. Chaos 31, 011101 (2021)
14. Zhang S., Li C., Zheng J., Wang X., Zeng Z., Peng X.: Generating any number of initial offset-booted coexisting cha’a’s double-scroll attractors via piecewise-nonlinear memristor. IEEE Trans. Ind. Electron. page Early Access, (2021)
15. Gu, J., Li, C., Chen, Y., Iu, H.H.C., Lei, T.: A conditional symmetric memristive system with infinitely many chaotic attractors. IEEE Access 8, 12394–12401 (2020)
16. Yang, Y., Wang, L., Duan, S., Luo, L.: Dynamical analysis and image encryption application of a novel memristive hyperchaotic system. Opt. Laser Technol. 133, 106553 (2021)
17. Pham, V.-T., Volos, C., Fortuna, L.: Memristor-based systems: nonlinearity, dynamics and applications. Eur. Phys. J. Spec. Top. 228, 1902–1906 (2019)
18. He, S., Sun, K., Peng, Y., Wang, L.: Modeling of discrete fracmemristor and its application. AIP Adv. 10, 015332 (2020)
19. Peng, Y., He, S., Sun, K.: A higher dimensional chaotic map with discrete memristor. AEU-Int. J. Electron. Commun. 129, 153539 (2021)
20. Peng, Y., He, S., Sun, K.: Chaos in the discrete memristor-based system with fractional-order difference. Results Phys. 24, 104106 (2021)
21. Bao, B., Li, H., Huagan, W., Zhang, X., Chen, M.: Hyperchaos in a second-order discrete memristor-based map model. Electron. Lett. 56, 769–770 (2020)
22. Li, H., Hua, Z., Bao, H., Chen, M., Bao, B.: Two-dimensional memristive hyperchaotic maps and application in secure communication. IEEE Trans. Ind. Electron. 68, 9931–9940 (2021)

23. Bao, B., Rong, K., Li, H., Li, K., Hua, Z., Bao, B.: Memristor-coupled logistic hyperchaotic map. IEEE Trans. Circuits Syst. II: Express Briefs, page Early Access (2021)

24. Bao, H., Hua, Z., Chen, M., Bao, B.: Discrete memristor hyperchaotic maps. IEEE Trans. Circuits Syst. I: Regular Papers, page Early Access (2021)

25. Li, K., Bao, H., Li, H., Ma, J., Hua, Z., Bao, B.: Memristive rulkov neuron model with magnetic induction effects. IEEE Trans. Ind. Inf., page Early Access, (2021)

26. Kong, S., Li, C., He, S., Cicek, S., Lai, Q.: A memristive map with coexisting chaos and hyperchaos. Chin. Phys. B, page Early Access. (2021)

27. Pecora, L.M., Carroll, T.L.: Synchronization of chaotic systems. Chaos 25, 097611 (2015)

28. Yousri, D.A., AbdelAty, A.M., Said, L.A., Elwakil, A.S., Maundy, B., Radwan, A.G.: Parameter identification of fractional-order chaotic systems using different metaheuristic optimization algorithms. Nonlinear Dyn. 95, 2491–2542 (2019)

29. Wei, D., Miao, Q., Tong, L., Tang, Y.: Identification of fractional-order systems with unknown initial values and structure. Phys. Lett. A 381, 1943–1949 (2017)

30. Peng, Y., Sun, K., He, S., Yang, X.: Parameter estimation of a complex chaotic system with unknown initial values. Eur. Phys. J. Plus 133, 305 (2018)

31. Chen, F., Ding, Z., Zhongrong, L., Zeng, X.: Parameters identification for chaotic systems based on a modified jaya algorithm. Nonlinear Dyn. 94, 2307–2326 (2018)

32. Gupta, S., Upadhyaya, V., Singh, A., Varshney, P., Srivastava, S.: Modeling of fractional order chaotic systems using artificial bee colony optimization and ant colony optimization. J. Intell. Fuzzy Syst. 35, 5337–5344 (2018)

33. Zhang, P., Yang, R., Yang, R., Ren, G., Yang, X., et al.: Parameter estimation for fractional-order chaotic systems by improved bird swarm optimization algorithm. Int. J. Modern Phys. C 30, 1950086 (2019)

34. Ouannas, A., Khennaoui, A.-A., Grassi, G., Bendoukhd, S.: On chaos in the fractional-order grassi-miller map and its control. J. Comput. Appl. Math. 358, 293–305 (2019)

35. Peng, Y., Sun, K., He, S., Peng, D.: Parameter identification of fractional-order discrete chaotic systems. Entropy 21, 27 (2019)

36. Peng, Y., Sun, K., He, S.: Dynamics analysis of chaotic maps: From perspective on parameter estimation by metaheuristic algorithm. Chin. Phys. B 29, 030502 (2020)

37. Jafari, S., Ahmadi, A., Panahi, S., Rajagopal, K.: Extreme multi-stability: When imperfection changes quality. Chaos Solitons & Fractals 108, 182–186 (2018)

38. Quaranta, G., Lacarbonara, W., Masri, S.F.: A review on computational intelligence for identification of nonlinear dynamical systems. Nonlinear Dyn. 99, 1709–1761 (2020)

39. Jafari, S., Sprott, J.C., Pham, V.-T., Golpayegani, S.M.R.H., Jafari, A.H.: A new cost function for parameter estimation of chaotic systems using return maps as fingerprints. Int. J. Bifurc. Chaos 24, 1450134 (2014)

40. Peng, Y., Sun, K., He, S.: An improved return maps method for parameter estimation of chaotic systems. Int. J. Bifurc. Chaos 30, 2050058 (2020)

41. Storn, R.M., Price, K.: Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. ICSI Technical Report TR-95-012., (1995)

42. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evolut. Comput. 1, 67–82 (1997)

43. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evolut. Comput. 15, 4–31 (2011)

44. Wang, D., Tan, D., Liu, L.: Particle swarm optimization algorithm: an overview. Soft Comput. 22, 387–408 (2018)

45. Rao, R.V.: Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. 7, 19–34 (2016)

46. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Appl. Math. Comput. 214, 108–132 (2009)

47. Meng, X., Gao, X.Z., Lu, L., Liu, Y., Zhang, H.: A new bio-inspired optimisation algorithm. J. Experimen. Theore. Artif. Intell. 28, 673–687 (2016)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.