IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 20, 2017, accepted August 4, 2017, date of publication August 11, 2017, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2738006

An Effective Hybrid Cuckoo Search Algorithm for Unknown Parameters and Time Delays Estimation of Chaotic Systems

JIAMIN WEI AND YONGGUANG YU
Department of Mathematics, Beijing Jiaotong University, Beijing 100044, China

Corresponding author: Yongguang Yu (ygyu@bjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772063 and in part by the Fundamental Research Funds for the Central Universities under Grant 2016JBM070.

ABSTRACT Parameter estimation is an important issue in nonlinear science, which can be formulated as a multi-dimensional problem. Numbers of nature-inspired meta-heuristic algorithms have been applied for parameter estimation of chaotic systems; however, many of them are not able to achieve an appropriate trade-off between exploration and exploitation. Therefore, this paper proposes an effective hybrid cuckoo search (HCS) algorithm to obtain higher quality solutions and convergence speed. Inspired by the powerful efficiency of differential evolution, the proposed HCS provides two novel mutation strategies to fully exploit the neighborhood among the current population. Furthermore, a crossover operator under self-adaptive parameters control is introduced to balance the exploration and exploitation ability of the proposed two mutation strategies. Besides, the opposition-based learning is incorporated into HCS for initializing population and producing new candidate solutions during the evolutionary process. HCS is further employed to estimate the unknown parameters and time delays of chaotic systems. Numerical simulations and comparisons with some other optimization methods are conducted on three chaotic systems with and without time delays to demonstrate the performance of HCS. The experimental results show a superiority of HCS in parameter estimation of chaotic systems, and can be regarded as a promising method in terms of its high calculation accuracy, fast convergence speed, and strong robustness.

INDEX TERMS Parameter estimation, chaotic systems, cuckoo search algorithm, hybrid algorithm, time delays.

I. INTRODUCTION

As a particular case of nonlinear dynamics, chaos is characterized by an unstable dynamic behavior that exhibits sensitive dependence on the initial condition and includes infinite unstable periodic motions. Because of the specific properties, chaos has been used in many academic and engineering fields of chemical reactions, power converters, secure communications, information processing, biological systems and mechanical systems and so on [1]–[6]. In recent years, many nonlinear systems have been proven to exhibit phenomenon of chaos. In particular, systems with time delays exhibit more complex and adequate dynamic behavior than those free of time delays [7]. Meanwhile, much attention has been gained for the study of control and synchronization of chaotic systems for its potential applications [8]–[10]. Many methods have been proposed to control and synchronize chaotic systems, only with a condition that parameters of chaotic systems are known in advance. However, in the real world such as secret communications, due to the complexity of chaotic systems, the exact true values of parameters are difficult to determine. Therefore, parameter estimation of chaotic systems has become an important issue of nonlinear science.

In the past decade, a lot of approaches have been put forward for solving the problem of parameter estimation of chaotic systems, which can be mainly classified as two basic methods, one is the synchronization method and the other is the optimization method. Chaos synchronization has attracted increasing interest since the pioneering work of Pecora and Carroll [11] in 1990, and the synchronization method is based on the stability analysis of chaotic systems and the control methods. This concept was firstly proposed by Parlitz [12], [13], and has been extensively investigated in many literatures concerning the parameter estimation of uncertain chaotic systems [14]–[17], but the design of both the controller and the updating law of parameter estimation is still a hard task in terms of the techniques and sensitivities depending on the considered system [18]. On the other hand, the gradient-based optimization techniques, especially those using artificial intelligence, have played an important role on the analysis and parameter estimation of nonlinear dynamic systems. In the second method, the unknown parameters are considered as a series of independent variables and parameter estimation is converted to a multi-dimensional optimization problem. Compared with the first method, the optimization method is not sensitive to the considered systems and easy to implement, thus, it is more applicable.

Recently, nature-inspired meta-heuristic algorithms, especially combined with stochastic search techniques, seem to be a more hopeful approach and provide a powerful means to solve the nonlinear optimization problems. These algorithms can be regarded as a promising alternative to the traditional gradient-based techniques, since they do not rely on any assumptions such as differentiability or continuity. As a matter of fact, heuristic algorithms depend only on the objective function to guide the search. Owing to these outstanding characteristics, different kinds of heuristic algorithms have been applied to estimate the unknown parameters of chaotic systems, including genetic algorithm (GA) [19], particle swarm optimization (PSO) [20], differential evolution (DE) [21], cuckoo search (CS) algorithm [22], artificial bee colony (ABC) [23] and so on [24]–[27]. Among the latest developed algorithms, cuckoo search (CS) is a population-based heuristic evolutionary technique proposed by Yang and Deb [22], [28], the basic idea of which comes out from the parasitic brood swarm intelligence technique in cuckoo species together with the Lévy flight behavior of some birds and fruit flies. Due to simple concepts, ease of implementation and few parameters, CS has attracted a great interest of researchers and been successfully applied in a variety of problems from diverse fields [29], [30]. However, CS is weak in exploiting the solutions and easily trapped into a local optimum. Thus, it is necessary to improve the performance of CS to obtain higher quality solutions and convergence speed. Many variations of CS have been proposed in recent years for solving function optimization problems [31]–[33]. Nevertheless, there is no specific algorithm to achieve the best solution for all optimization problems [34]. Meanwhile, it is hard to achieve an appropriate trade-off of CS between exploration and exploitation [35]. In particular, it is common to find that CS shows relatively slow convergence speed when applied to parameter estimation of chaotic systems.

In the basic CS algorithm, there are two main phases of generating new solutions, including the exploration phase for generation of new eggs via Lévy flights and the exploitation phase for generation of new eggs via replacement of a fraction of eggs. It is noteworthy that a lot of work have been done to improve the first phase, such as self-adaptation adjustment of parameters, selection of variable step size and combination of other searching mechanisms [36]–[39]. However, researches devoted to improve the second phase of biased/selective random walk (BSRW) are still not sufficient so far. Therefore, it seems necessary to put forward new improved or enhanced techniques to ameliorate the quality of solutions and convergence speed for BSRW. Due to the selection of the simple random walk rule, the second phase significantly lacks of diversity of solutions and the local search capability is relatively weak, which may decrease the quality of optimization. Moreover, if the searching environment is complex with numerous local optima, the solutions may get trapped in a local optimum. Motivated by this aspect, we introduce an effective hybrid cuckoo search (HCS) algorithm to further enhance the exploration and the exploitation ability of the basic CS. To be specific, an improved differential evolution (IDE) strategy is introduced to the basic CS algorithm to discourage premature convergence and increase the exploitation ability of local search. In IDE, a new mutation operator based on two searching schemes under adaptive parameters control is put forward so that the discovered nests can be rebuilt by making full use of the current individual's information in a reasonable way. Besides, the opposition-based learning (OBL) is incorporated into HCS for initializing population and producing new candidate solutions during the evolutionary process. By considering an estimate and its corresponding estimate simultaneously, OBL can provide a faster convergence rate and a higher chance of finding candidate solutions closer to the global optimum. The HCS algorithm is further applied to estimate the unknown parameters and time delays of chaotic systems. Numerical simulations are performed on several chaotic systems, and statistically compared with some typical existing approaches. All of the simulation results demonstrate the effectiveness and robustness of HCS, and its superiority to the other comparative algorithms.

The rest of this paper is organized as follows. In Section II, the problem of parameter estimation of both chaotic systems with and without time delays is formulated from the viewpoint of optimization, respectively. In Section III, the HCS algorithm is proposed in sufficient details after a brief review of the basic CS. In Section IV, simulations and comparisons with some existing approaches are done on three typical chaotic systems, and results analyses together with discussions are provided as well. Finally, conclusions are drawn in Section V.

II. PROBLEM FORMULATION

A. FOR CHAOTIC SYSTEMS WITHOUT TIME DELAYS

Consider the following n-dimensional chaotic system described by ordinary differential equation (ODE):

$$\dot{X}(t) = F(X(t), X_0, \theta), \tag{1}$$

where $X(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T \in R^n$ denotes the state vector, $X_0 = (x_{10}, x_{20}, \ldots, x_{n0})^T$ denotes the initial state and $\theta = (\theta_1, \theta_2, \ldots, \theta_d)^T$ is a set of original parameters.

Suppose the structure of system (1) is known in advance, then the estimated system can be written as:

$$\dot{\tilde{X}}(t) = F(\tilde{X}(t), X_0, \tilde{\theta}), \tag{2}$$

where $\tilde{X}(t) = (\tilde{x}_1(t), \tilde{x}_2(t), \ldots, \tilde{x}_n(t))^T \in R^n$ is the state vector of the estimated system, $\tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2 \ldots, \tilde{\theta}_d)^T$ is a set of systematic parameters to be estimated.

To estimate the unknown parameters, the following objective function is defined as:

$$F = \frac{1}{N}\sum_{k=1}^{N} \|X_k - \tilde{X}_k\|^2, \tag{3}$$

where $k = 1, 2, \ldots, N$ is the sampling time point and N denotes the length of data used for parameter estimation, X_k and $\tilde{X}_k$ $(k = 1, 2, \ldots, N)$ denote the state vector of the original and the estimated system at time k, respectively. The parameter estimation of system (1) can be achieved by searching suitable $\tilde{\theta}$ such that the objective function (3) is minimized, i.e,

$$\theta^* = \arg\min_{\tilde{\theta}\in\Theta} F, \tag{4}$$

where Θ is the searching space admitted for parameters.

B. FOR CHAOTIC SYSTEMS WITH TIME DELAYS

Consider the following n-dimensional chaotic system described by delay differential equation (DDE):

$$\dot{X}(t) = F(X(t), X(t-\tau), X_0, \theta), \tag{5}$$

where $X(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T \in R^n$ denotes the state vector, $X_0 = (x_{10}, x_{20}, \ldots, x_{n0})^T$ denotes the initial state for $t \leq \tau$ and $\theta = (\theta_1, \theta_2, \ldots, \theta_d)^T$ is a set of original parameters. The time delay τ is treated as a parameter to be estimated as well in this paper.

Suppose the structure of system (5) is known in advance, then the estimated system can be written as:

$$\dot{\tilde{X}}(t) = F(\tilde{X}(t), \tilde{X}(t-\tilde{\tau}), X_0, \tilde{\theta}), \tag{6}$$

where $\tilde{X}(t) = (\tilde{x}_1(t), \tilde{x}_2(t), \ldots, \tilde{x}_n(t))^T \in R^n$ is the state vector of the estimated system, $\tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_d)^T$ and $\tilde{\tau}$ are parameters to be estimated.

The objective function for this case is constructed as similar as formula (3). Then the parameter estimation of system (5) can be achieved by searching suitable $\tilde{\theta}$ such that the objective function is minimized, i.e,

$$(\theta^*, \tau^*) = \arg\min_{(\tilde{\theta},\tilde{\tau})\in\Theta} F, \tag{7}$$

where Θ is the searching space admitted for parameters and time delays.

Obviously, it is not easy to estimate the unknown parameters and time delays of chaotic systems because of its dynamic instability. Moreover, it can be seen that the objective function is a multi-dimensional nonlinear function with multiple local optimum, and it is difficult to obtain the global optimal solution effectively and accurately using traditional optimization methods. In this paper, an effective hybrid cuckoo search algorithm is proposed and applied to the problem of parameter estimation of chaotic systems with and without time delays.

III. HYBRID CUCKOO SEARCH OPTIMIZATION ALGORITHM

In this section, an effective hybrid cuckoo search (HCS) algorithm is proposed in order to further enhance the exploration and the exploitation ability of the basic CS. To be specific, an improved differential evolution (IDE) strategy under adaptive parameters control is introduced to the second phase of CS to discourage premature convergence and increase the exploitation ability of local search. On the other hand, the opposition-based learning (OBL) is incorporated into HCS for initializing population and producing new candidate solutions in evolutionary generations, which can guide the population toward the more promising areas and spread it as much as possible over the searching space. The main strategies of HCS are described in the following subsections after a brief review of the basic CS algorithm.

A. CUCKOO SEARCH ALGORITHM

CS algorithm is a simple yet very promising stochastic population-based method. For simplicity in describing the basic CS algorithm, three idealized rules [22] are used: (1) Each cuckoo bird lays one egg at a time and dumps it at a random chosen host nest; (2) The best nests with high-quality eggs will be carried over to the next generations; (3) The number of available host nests is fixed, and the host bird may discover the alien egg laid by a cuckoo with a probability $P_a \in [0, 1]$.

In CS, each egg in a nest represents a solution, and each cuckoo is supposed to lay only one egg (thus representing one solution), the aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests [40]. CS uses a balanced combination of a local random walk and a global explorative random walk, which are controlled by a switching parameter P_a. After each random walk, a greedy strategy is used to select better solutions from the current and new generated solutions according to their fitness values.

Based on the rules and description above, the global random walk is carried out by using Lévy flights as follows

$$X_i^{t+1} = X_i^t + \alpha \oplus \text{Lévy}(\lambda), \tag{8}$$

where α $(\alpha > 0)$ is the step size related to the optimization problem scale, and the product $\oplus$ denotes the entry-wise multiplication. Lévy flights essentially provide a random walk, the random steps of which are drawn from a Lévy distribution for large steps:

$$Lévy \sim t^{-\lambda}, \quad (1 < \lambda \leq 3), \tag{9}$$

which has an infinite variance with an infinite mean.

After Lévy flights random walk, CS continues to generate new solutions in terms of biased/selective random walk which employs a crossover operator. Considering the probability of cuckoos being discovered, a crossover operator is used to construct a new solution:

$$X_i^{'} = \begin{cases} X_i + r \cdot (X_{r1} - X_{r2}), & \text{if } (rand[0,1] > P_a), \\ X_i, & \text{otherwise}, \end{cases} \tag{10}$$

where $r1$ and $r2$ are mutually different random integers; r denotes the scaling factor which is a uniformly distributed random number in the interval [0, 1]. By using the greedy strategy, the next generation solution is selected from X_i and $X_i^{'}$ according to their fitness values. At the end of each iteration process, the best solution obtained so far is updated.

The procedures of basic CS algorithm can be described as the pseudo code shown in Algorithm 1.

Algorithm 1 Pseudo Code of the Basic CS Algorithm

```
Generate an initial population of N host nests X_i, (i = 1, 2, ..., N);
Evaluate the fitness value of each nest X_i;
Determine the best nest with the best fitness value;
while t ≤ MaxGeneration do
    for i = 1, 2, ..., N do
        Generate a cuckoo X_i randomly by Lévy flights according to Eq. (8);
        Evaluate the fitness value F_i = f(X_i);
        Choose a random nest X_j;
        if (F_j < F_i) then
            Replace nest X_i with X_j;
        end if
    end for
    Abandon a fraction P_a of worse nests and build new ones according to Eq. (10);
    Keep the best nest with quality solution;
    Rank the nests and find the current best one;
    Pass the current best nest to the next generation;
end while
```

B. DIFFERENTIAL EVOLUTION-BASED RANDOM WALK

The efficacy of solving optimization problems using evolutionary algorithms relies on the choice of strategies that are used to generate new individuals and their associated parameters [41]. In this section, an improved differential evolution (IDE) strategy under adaptive parameters control is introduced to the second phase of CS, and used instead of the simple random walk in the basic CS algorithm.

DE has strong global search capability and shows pretty good convergence ability. Many schemes have been proposed based on different mutation strategies, where the notation "DE/x/y/z" is used to denote the strategies. "DE" stands for differential evolution, "x" represents the base vector to be perturbed, "y" is the number of differential vectors considered for perturbation, and "z" denotes the type of crossover scheme (exp: exponential; bin: binomial). There are several mutation strategies which are also frequently used in the literature [42]:

$$\text{DE/rand/1: } V_i = X_{r1} + F \cdot (X_{r2} - X_{r3}), \tag{11}$$

$$\text{DE/best/1: } V_i = X_{best} + F \cdot (X_{r1} - X_{r2}), \tag{12}$$

$$\text{DE/current-to-best/1: } V_i = X_i + F \cdot (X_{best} - X_i) + F \cdot (X_{r1} - X_{r2}), \tag{13}$$

where X_{best} is the best individual in the current generation; r_1, r_2 and r_3 are mutually exclusive integers randomly chosen from the range $[1, N]$, and must be different from the base index i; F is the mutation factor. The selection of different mutation strategies has great impact on the optimization performance.

Inspired by the mutation strategy "DE/best/1" and "DE/current-to-best/1", a new mutation operator based on two searching schemes under adaptive parameters control is proposed as follows:

$$\text{CS/best/1: } V_i = X_{best} + F_1 \cdot (X_{r1} - X_{r2}), \tag{14}$$

$$\text{CS/current-to-best/1: } V_i = X_i + F_1 \cdot (X_{best} - X_i) + F_2 \cdot (X_{r1} - X_{r2}), \tag{15}$$

where F_1, F_2 are two independent mutation factors.

"CS/best/1" is able to explore the region around the best solution in the current population. A crossover operator is incorporated to "CS/best/1" which can be expressed as:

$$V_{i,j} = \begin{cases} X_{best,j} + F_1 \cdot (X_{r1,j} - X_{r2,j}), \\ \qquad \text{if } (rand[0,1] > P_a), \\ X_{best,j}, \quad \text{otherwise}. \end{cases} \tag{16}$$

This strategy is good at increasing the convergence speed and population diversity, but is easy to fall into local optimum.

"CS/current-to-best/1" affords two difference vectors to perturb the target vector and generates a new solution. Regard with the discovering probability, then "CS/current-to-best/1" directed random walk can be expressed as:

$$W_{i,j} = \begin{cases} X_{i,j} + F_1 \cdot (X_{best,j} - X_{r1,j}) + F_2 \cdot (X_{r2,j} - X_{r3,j}), \\ \qquad \text{if } (rand[0,1] > P_a), \\ X_{i,j}, \quad \text{otherwise}. \end{cases} \tag{17}$$

It maintains the population diversity and the global search capability at the same time. Besides, it favors exploitation since the new individual is obtained by considering both of the current best vector and random vectors in the neighborhood.

Considering the properties of the two searching schemes, a crossover operator under self-adaptive parameters control is incorporated to the IDE strategy, to keep a good balance between exploitation and exploration. The crossover operation is given as follows:

$$U_{i,j}^{t+1} = \begin{cases} V_{i,j}, & \text{if } (rand[0,1] \le \text{CR}) \text{ or } (j = j_{rand}), \\ W_{i,j}, & \text{otherwise}, \end{cases} \tag{18}$$

where CR is a self-adaptive crossover rate; j_{rand} is a randomly chosen integer in the range of $[1, D]$. CR decides the fraction of elements copied from the mutant vectors, thus it is of great importance in balancing the local and global search processes. In implementation, the self-adaptive CR is set as follows:

$$CR = \begin{cases} CR_1 + (CR_2 - CR_1) * \dfrac{max(f) - f(i)}{max(f) - min(f)}, \\ \qquad \text{if } (f(i) \leq mean(f)), \\ CR_1, \quad \text{otherwise}, \end{cases} \tag{19}$$

where $mean(f)$, $min(f)$ and $max(f)$ denotes the mean, the minimum and the maximum fitness value of the current population, respectively; CR_1, CR_2 are two predefined parameters.

C. OPPOSITION-BASED LEARNING

Opposition-based learning (OBL), introduced by Ti-zhoosh [43], is a new concept in computational intelligence. The main idea behind OBL is to consider both of a solution and its corresponding opposite solution in order to get a better approximation of the current candidate solutions. It has been proven to be an effective method to enhance various optimization approaches [18], [25]. Hence, the OBL idea is incorporated into our proposed algorithm, to further increase diversity and speed up the convergence.

Suppose $X = (x_1, x_2, \ldots, x_n)$ is a solution in an n-dimensional space, where $x_i \in [Lx_i, Ux_i]$, $(i = 1, 2, \ldots, n)$. The opposite solution $X' = (x_1', x_2', \ldots, x_n')$ is given by:

$$x_i' = Lx_i + Ux_i - x_i. \tag{20}$$

Let $f(\cdot)$ be a fitness function via which the fitness value can be evaluated. According to the above given definitions of X and X', if $f(X') \leq f(X)$, then X is replaced with X', otherwise X is kept. Thereby, the solution and its opposite solution are evaluated simultaneously in order to obtain the fitter one. OBL is implemented to initialize population and produce new solutions during evolution process.

D. THE MAIN PROCEDURE OF HCS

In this paper, an effective HCS algorithm is proposed based on the strategies described above. Firstly, OBL is used in population initialization to obtain fitter starting candidate solutions and increase the diversity. Then, to discourage premature convergence and increase the exploitation ability of local search, the proposed IDE under adaptive parameters control strategy is embedded into the second phase of CS, instead of the simple random walk in CS. When the procedures of the proposed modified CS algorithm are finished, populations are updated based on opposition-based generation jumping for further increasing the diversity and accelerating convergence during the evolutionary process. The main procedure of HCS is given in Algorithm 2. The proposed approaches in this paper are identified with boldface.

Several new CS variants combined with the DE strategy have been put forward to address the global optimization problems [44], [45], and our improved differential evolution (IDE) strategy under adaptive parameters control seems to share similarities with them. However, the proposed strategy has its own specific characteristics, which makes it different from the DE techniques used in the literatures, as clarified below.

1) Two mutation strategies are adopted in IDE (namely "DE/best/1" and "DE/current-to-best/1"), while DECS presented in [44] only employs the simple "DE/rand/1" strategy which offers lower diversity to search for new solutions.
2) The searching schemes in the BSRW stage proposed in our HCS (namely "CS/best/1" and "CS/current-to-best/1") are distinct from "CS/rand/1" and "CS/best/2" raised in SACS [45]. Compared with the random scale factor φ defined in SACS, two independent mutation factors are put to use in IDE. Moreover, a crossover operator under self-adaptive parameters control is incorporated to the IDE strategy according to the properties of the two searching schemes, which is expected to keep a good balance between exploitation and exploration.
3) In implementation, the crossover rate CR of IDE is set self-adaptive, whereas SACS utilizes a self-adaptive parameter P_a. Besides, DECS is just coupled with differential evolution but in the absence of parameter adaptation approach.

Algorithm 2 Pseudo Code of HCS Algorithm

Initialization via OBL;
Evaluate the fitness value of each nest through the objective function Eq. (3);
Determine the best nest with the best fitness value;
while t $\leq$ MaxGeneration **do**
  for $i = 1, 2, \ldots, N$ **do**
    Generate a cuckoo X_i randomly by Lévy flights according to Eq. (8);
    Evaluate the fitness value $F_i = f(X_i)$;
    Choose a random nest X_j;
    if $(F_j < F_i)$ **then**
      Replace nest X_i with X_j;
    end if
  end for
  Determine the best nest from the current fitness values;
  for $i = 1, 2, \ldots, N$ **do**
    Calculate **the self-adaptive crossover rate CR** using Eq. (19);
    Search for a new solution using **IDE strategy** using Eqs. (16-18);
  end for
  Opposition-based generation jumping;
  Keep the best nest with quality solution;
  Rank the nests and find the current best one;
  Pass the current best nest to the next generation;
end while

TABLE 1. Simulation results of different methods for Lorenz system (21).

Method	CS	ACS	NNCS-F	SACS	HCS
a	9.999424028902222	10.000000071441821	9.998979914774930	9.999934944684826	9.999999999999330
$\frac{\|a-10\|}{10}$	5.76E-05	7.14E-09	1.02E-04	6.51E-06	**6.70E-14**
b	28.003436851037556	28.000000793966120	28.000862369020592	28.000098949286365	27.999999999999993
$\frac{\|b-28\|}{28}$	1.23E-04	2.84E-08	3.08E-05	3.53E-06	**2.54E-16**
c	2.665792245110993	2.666666529739050	2.666353328596084	2.666687471091286	2.666666666666795
$\frac{\|c-8/3\|}{8/3}$	3.64E-05	5.13E-08	1.18E-04	7.80E-06	**5.37E-15**
F	1.21E-03	9.12E-07	4.40E-04	7.26E-05	**4.70E-13**

On the other hand, although OBL has been introduced into CS, our method owns its distinguishing features as similar done in [25] at each iteration process. For example, in [46], OBL together with the best solution is merged in the LFRW phase rather than BSRW. And for NCS [47], modifications are based on the generalized opposition-based learning (GOBL) instead of OBL used in this paper.

In summary, the proposed HCS algorithm exhibits advantages and differences from other methods. Furthermore, extensive simulations are conducted in the following section to validate its effectiveness and efficiency.

E. COMPUTATIONAL COMPLEXITY OF HCS

Compared with the original CS algorithm, HCS needs to perform additional computations on the IDE strategy and OBL process. During one generation, the mutation operations of the IDE strategy are executed before crossover, the average complexity of this process is $O(N \cdot n)$. Then, the crossover operator under adaptive parameters control is calculated to obtain new solutions in BSRW, the complexity of which is $O(N \cdot \log(N))$. In addition, calculations of opposition population and population sorting continue to be carried out. The computation complexity of these two procedures takes $O(2\cdot N\cdot n)+O(N\cdot\log(N))$. Since the complexity of the original CS algorithm is $O(Gmax \cdot N \cdot n)$ where $Gmax$ is the maximal number of generation, the total computational complexity of HCS is $O(Gmax\cdot[N\cdot n+N\cdot\log(N)+2\cdot N\cdot n+N\cdot\log(N)])$, which is simplified to $O(Gmax \cdot N \cdot n)$. Hence, the proposed HCS does not significantly increase the overall complexity compared with the original CS.

IV. SIMULATIONS RESULTS

To demonstrate the performance of the proposed scheme, three typical chaotic systems with and without time delays are taken for example in this section. The simulations are performed using MATLAB 7.1 on Intel Core® i5-3380, 2.90GHz with 4GB RAM. The original system evolves freely from a random initial state, and randomly select a state as the initial state X_0 for parameter estimation after a period of transient process. To achieve a fine balance between the performance of algorithms and having enough sample data for credibility, the number of states of the original and estimated system for calculating the objective function (3) is set to 300 according to the existing simulations In the simulation, the parameters of the proposed HCS algorithm are set as follows: the population size $N = 40$, the maximum iteration number $M = 50$, the probability of discovering an alien egg $P_a = 0.25$ which is suggested in [22], mutation factors and adaptive crossover rates are set to $F_1 = 0.6$, $F_2 = 0.01$ and $CR_1 = 0.1$, $CR_2 = 0.6$, respectively, according to the extensive experiments. In order to evaluate the effectiveness and efficiency of HCS, comparisons with the classical CS algorithm, some CS improved variants and state-of-the-art algorithms i.e., ACS [48], NNCS-F [49], SACS [45], ABC [23], DE [50], PSO [7], OSOA [25] and HABC [18] are carried out on different systems. Parameter settings of these compared algorithms in our experiments are the same as recommended in their original papers. For a fair comparison, the same computation effort is used in each compared algorithm, i.e., the number of running times, population size, maximum iteration number, length of sampling time points, step size, and searching ranges of parameters. For each problem, the experimental results are averaged via 20 independent runs.

A. SIMULATIONS ON CHAOTIC SYSTEMS WITHOUT TIME DELAYS

In this subsection, we consider two typical chaotic systems without time delays, namely, Lorenz system and Rössler system. Simulations are conducted in order to show how the proposed HCS algorithm can improve the performance of CS, and at the same time to validate its competitiveness upon other efficient methods. For the first system, the proposed approach is compared with three improved CS variants (namely ACS, NNCS-F and SACS) besides classical CS. Regarding to the second system, comparison with some other state-of-the-art algorithms (namely ABC, CS, DE and PSO) are performed to further test the effectiveness of HCS.

1) COMPARISON WITH OTHER CS IMPROVED VARIANTS

Example 1: Consider the following Lorenz system [51] described by

$$\begin{cases} \dot{x} = a(y - x), \\ \dot{y} = x(b - z) - y, \\ \dot{z} = xy - cz, \end{cases} \tag{21}$$

where a, b and c are unknown parameters to be estimated. The original system is assigned with true parameters: $a = 10$, $b = 28$ and $c = 8/3$, under which the system is chaotic.

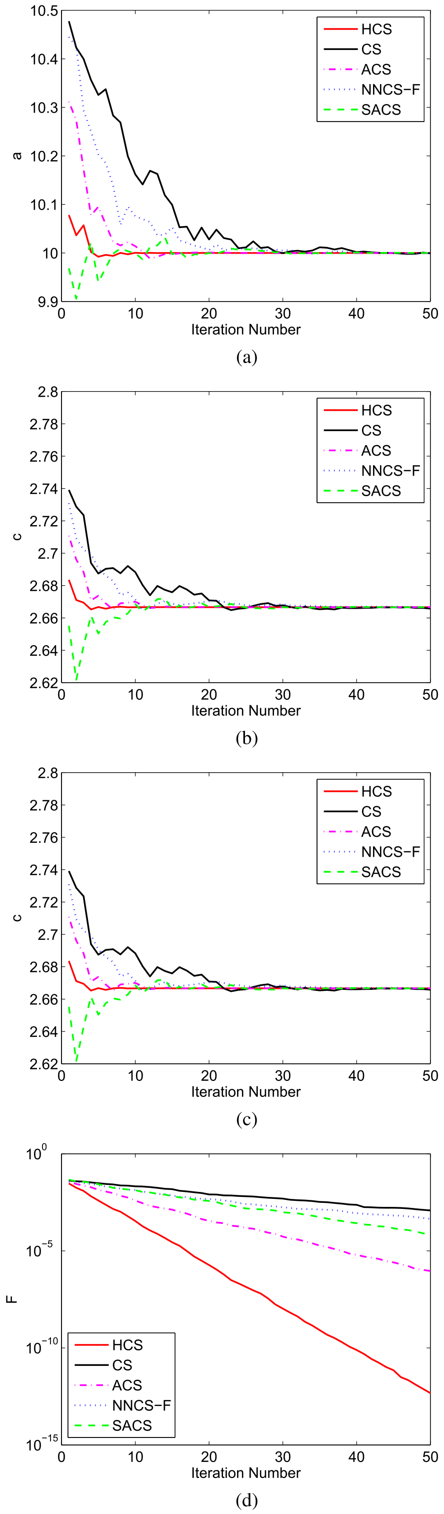

FIGURE 1. Evolution process of the estimated parameters and the objective function values for Lorenz system (21). (a) Evolution process of *a*. (b) Evolution process of *b*. (c) Evolution process of *c*. (d) Evolution process of *F*.

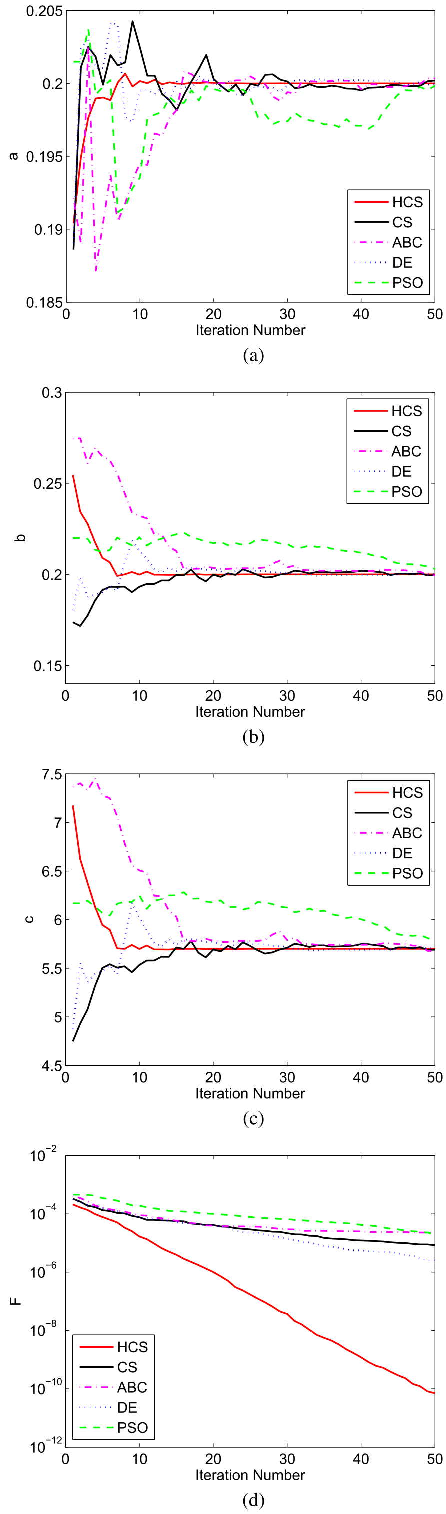

FIGURE 2. Evolution process of the estimated parameters and the objective function values for Rössler system (22). (a) Evolution process of *a*. (b) Evolution process of *b*. (c) Evolution process of *c*. (d) Evolution process of *F*.

The searching ranges of unknown parameters are predefined in $\Omega = [9, 11] \times [20, 30] \times [2, 3]$.

The statistical results of the mean estimated values with corresponding relative error values and the objective function values are listed in Table 1. It can be seen that the estimated values obtained by HCS are closer to the true parameter values than those by the basic CS, ACS, NNCS-F and SACS algorithm. Especially, from the relative error values

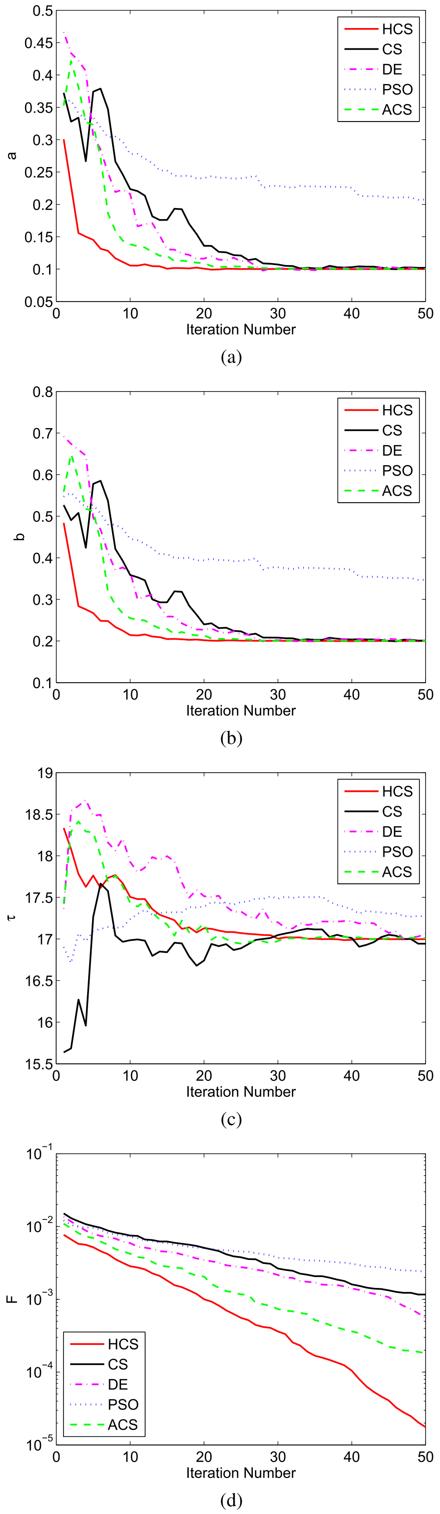

FIGURE 3. **Evolution process of the estimated parameters and the objective function values for time-delay Mackey-Glass chaotic system (23). (a) Evolution process of *a*. (b) Evolution process of *b*. (c) Evolution process of *c*. (d) Evolution process of *F*.**

marked in bold, it is obvious that the HCS algorithm is with higher precision than the other algorithms. Besides, the mean objective function values produced by HCS are also significantly better than those of the other three approaches.

The evolution process of the average results of the estimated parameters and the objective function values for system (21) are shown in Fig. 1. From the figure, it can be easily found the estimated parameters can converge to the matching true values at an earlier stage by HCS. We can observe that its ability to search for the true values of unknown parameters is successful. What's more, Fig. 1 shows that the objective function value generated by HCS decreases to zero much faster than the comparison algorithms, which implies that HCS can converge at a faster rate to the global optimal solution in identifying the unknown parameters of system (21). From the foregoing discussion, it can be concluded that the HCS algorithm contributes to superior performance in terms of efficiency, quality, and robustness.

2) COMPARISON WITH OTHER STATE-OF-THE-ART ALGORITHMS

Example 2: Consider the following Rössler system [52] described by

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + ay, \\ \dot{z} = b + xz - cz, \end{cases} \tag{22}$$

where a, b and c are unknown parameters to be estimated. The original system is assigned with true parameters: $a = 0.2$, $b = 0.2$ and $c = 5.7$, under which the system is chaotic. The searching ranges of unknown parameters are predefined in $\Omega = [0.01, 0.5] \times [0.01, 0.5] \times [2, 10]$.

For system (22), Table 2 shows the comparison results of the mean estimated values with corresponding relative error values and the objective function values by different methods over 20 independent runs. According to Table 2, it can be noted that HCS has more accurate results than the basic ABC, CS, DE and PSO algorithm, which indicates that HCS significantly outperforms the comparison algorithms in estimating parameters of system (22). Fig. 2 depicts the convergence process of the average results of the estimated parameters and objective function values. From the figure, it is obvious that HCS can converge to the optimal solution more rapidly and accurately than the other algorithms. Therefore, the HCS algorithm demonstrates the good performance in aspects of robustness and convergence accuracy, which is highly competitive with those of ABC, CS, DE and PSO for Rössler system.

B. SIMULATIONS ON CHAOTIC SYSTEMS WITH TIME DELAYS

For the problem of parameter estimation of chaotic systems with time delays, the identification results using HCS are compared with the basic CS, DE and PSO algorithm together with a single CS variant named ACS.

Example 3: Consider the following time-delay Mackey-Glass chaotic system [53] described by

$$\dot{x}(t) = -ax(t) + \frac{bx(t-\tau)}{1 + x(t-\tau)^{10}}, \tag{23}$$

TABLE 2. Simulation results of different methods for Rössler system (22).

Method	ABC	CS	DE	PSO	HCS
a	0.200332879145344	0.200093060642504	0.199987083134566	0.199736274579364	0.200000000148896
$\frac{\|a-0.2\|}{0.2}$	1.66E-03	9.88E-04	1.83E-04	7.67E-04	**8.90E-10**
b	0.199520027459480	0.200374005282876	0.199954320223319	0.200475128423103	0.200000000225163
$\frac{\|b-0.2\|}{0.2}$	2.40E-03	2.15E-03	9.79E-04	1.51E-02	**3.31E-08**
c	5.678580139199112	5.712176338344728	5.698462766945982	5.713696249783523	5.700000000132213
$\frac{\|c-5.7\|}{5.7}$	3.76E-03	1.45E-03	9.15E-04	1.38E-02	**2.86E-08**
F	2.27E-05	8.40E-06	2.52E-06	1.95E-05	**6.99E-11**

TABLE 3. Simulation results of different methods for time-delay Mackey-Glass system (23).

Method	CS	DE	PSO	ACS	HCS
a	0.101896336173199	0.100324669288395	0.207327381826193	0.100283551231219	0.100005588430890
$\frac{\|a-0.1\|}{0.1}$	1.90E-02	3.25E-03	1.07E+00	2.84E-03	**5.59E-05**
b	0.200945998814112	0.201344257226055	0.347186852712780	0.200634786816886	0.199989995924342
$\frac{\|b-0.2\|}{0.2}$	4.73E-03	6.72E-03	7.36E-01	3.17E-03	**5.00E-05**
τ	16.943115414975360	17.060382349648101	17.270742534325944	17.010623747704901	16.998554013823320
$\frac{\|\tau-17\|}{17}$	3.35E-03	3.55E-03	1.59E-02	6.25E-04	**8.51E-05**
F	1.16E-03	5.55E-04	2.41E-03	1.82E-04	**1.75E-05**

TABLE 4. Analysis of HCS features and comparisons.

System	CS	CS-IDE	CS-OBL	HCS	OSOA	HABC
Lorenz	1.21E-03	9.58E-08	3.58E-09	**4.70E-13**	2.89E-10	2.68E-06
Rössler	8.40E-06	1.15E-06	2.48E-10	**6.99E-11**	2.32E-06	3.15E-05
Mackey-Glass	1.16E-03	1.72E-04	1.31E-04	**1.75E-05**	1.28E-03	5.82E-03
Ranking	5.00	3.33	2.33	**1.00**	3.67	5.67

with a, b and τ are unknown parameters to be estimated. The original system is assigned with true parameters: $a = 0.1$, $b = 0.2$ and $\tau = 17$, under which the system is chaotic. The searching ranges of unknown parameters are predefined in $\Omega = [0.05, 1] \times [0.05, 1] \times [12, 20]$.

The statistical results including the mean estimated values, the relative error values and the objective function values via different methods over 20 independent runs are summarized in Table 3. In addition, the evolution process of the average results of the estimated parameters and the objective function values for time-delay chaotic system (23) are shown in Fig. 3.

Based on Fig. 3 and Table 3, it can be observed that all the algorithms have a certain capability of estimating parameters, but the performance of HCS is much better than the other four algorithms, and supplies more precise and robust results with faster convergence speed. In particular, the relative error values produced by HCS marked in bold are all smaller than those by the basic CS, DE and PSO and ACS algorithm; all the estimated parameters converge to the corresponding matching true values more rapidly via the HCS algorithm. In general, HCS is significantly better and statistically more robust than the listed comparison algorithms in terms of convergence precision and searching efficiency. Furthermore, it also can be concluded that HCS is very capable for parameter estimation of chaotic systems with time delays.

C. EFFECTIVENESS OF THE HCS FEATURES

The proposed HCS is mainly composed of two features, including differential evolution-based random walk and opposition-based learning. This section is intended to investigate the effectiveness of each strategy in HCS by sequential activation of these features. For this purpose, the following two variants of CS are considered:

1) CS employs the IDE strategy under adaptive parameters control, denoted as CS-IDE;
2) CS integrates with OBL, referred to CS-OBL.

The same parameter settings are applied in the aforementioned variants in implementation. Moreover, algorithms are compared with another two especial methods, namely OSOA [25] and HABC [18], which also adopt the OBL optimization tool in their literatures. In our experimental studies, the three examples for parameter estimation are set as benchmark functions with identical performance criteria. Besides that, the Friedman test [54] is conducted on the simulation results to obtain the reliable statistical conclusion.

Table 4 presents the statistical results of CS, CS-IDE, CS-OBL, HCS, OSOA and HABC, in terms of the mean objective function values. From Table 4, it can be seen that CS-IDE and CS-OBL yield significantly higher accurate solutions than the original CS algorithm. The combined effect

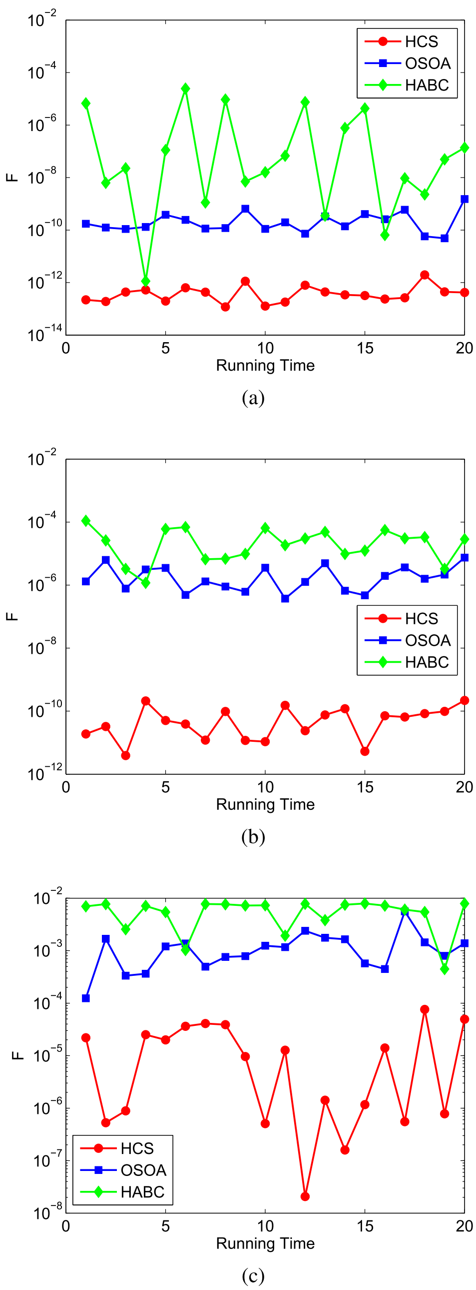

FIGURE 4. The objective function values of HCS, OSOA and HABC for systems (21-23) during 20 independent runs. (a) Results for Lorenz system (21). (b) Results for Rössler system (22). (c) Results for time-delay Mackey-Glass system (23).

of the two features, i.e. HCS, is also given in Table 4 and it shows that the proposed HCS algorithm is much more efficient in solving all the parameter estimation problems. Furthermore, with respect to the average ranking of the six algorithms in Table 4, it can be figured out that HCS achieves the overall best performance followed by CS-OBL, CS-IDE, OSOA, CS and HABC. This indicates that the OBL feature has bigger impact than the adaptive IDE strategy on the results of the HCS. Meanwhile, algorithms incorporating each feature demonstrate the ability to improve CS, and are even better than OSOA and HABC in optimization. HABC, a hybrid artificial bee colony algorithm, is proven to be good at handling parameter identification of uncertain fractional-order chaotic systems, however, shows weaker capacity for integer-order chaotic systems, which may be due to the insufficient population size for evolution. In addition, the objective function values of HCS, OSOA and HABC during 20 independent runs are plotted in Fig. 4, to further show the higher convergence accuracy of our proposed algorithm.

V. CONCLUSIONS

In this paper, an effective hybrid cuckoo search (HCS) algorithm is proposed to estimate the unknown parameters and time delays of chaotic systems from the perspective of optimization. The HCS algorithm is improved mainly in two aspects: differential evolution-based random walk and opposition-based learning (OBL). On the one hand, an improved differential evolution (IDE) strategy is introduced to the second phase of CS to discourage premature convergence and increase the exploitation ability of local search, instead of the simple random walk in the basic CS algorithm. On the other hand, the OBL is incorporated into HCS for population initialization and generation jumping in order to increase the diversity and accelerate convergence during the evolutionary process. Advantages of the proposed algorithm are easy to implement, fast convergence speed, and few parameters to adjust.

To verify the performance of the HCS algorithm, numerical simulations are performed on three typical chaotic systems, namely Lorenz, Rössler and time-delay Mackey-Glass chaotic system, by comparisons with some other CS improved variants and state-of-the-art methods besides the basic CS. The simulation results show that the HCS algorithm could estimate the unknown parameters and time delays of chaotic systems more rapidly, more accurately and more stably than the compared algorithms. Furthermore, the effectiveness of the two HCS features is investigated in details, respectively. It turns out that each feature has outstanding ability to improve the original CS algorithm, and the experimental results of them even significantly outperform some other compared algorithms when solving parameter estimation problems. In light of the results analyses and comparisons, it can be concluded that HCS is an effective, robust and promising algorithm for parameter estimation of uncertain both chaotic systems with and without time delays. Although this paper mainly concentrates on the parameter estimation of chaotic systems, we believe that the HCS algorithm will also be beneficial for the applications of synchronization and control of chaotic systems and various optimization problems. The future work is to further improve the performance of HCS through nonhomogeneous searching laws and dynamic sub-population sizes. Meanwhile, we also plan to apply the HCS algorithm for parameter estimation of chaotic systems with random noises or multistochastic disturbances, and some real-world optimization problems as well.

REFERENCES

[1] I. R. Epstein and J. A. Pojman, *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos*. Oxford, U.K.: Oxford Univ. Press, 1998.
[2] Z. T. Zhusubaliyev and E. Mosekilde, *Bifurcations and Chaos in Piecewise-Smooth Dynamical Systems: Applications to Power Converters, Relay and Pulse-Width Modulated Control Systems, and Human Decision-Making Behavior*. Singapore: World Scientific, 2003.
[3] S. Wang, J. Kuang, J. Li, Y. Luo, H. Lu, and G. Hu, "Chaos-based secure communications in a large community," *Phys. Rev. E*, vol. 66, no. 6, p. 065202, 2002.
[4] J. S. Nicolis, *Chaos and Information Processing: A Heuristic Outline*. Singapore: World Scientific, 1991.
[5] H. Degn, A. V. Holden, and L. F. Olsen, *Chaos in Biological Systems*, vol. 138. New York, NY, USA: Springer, 2013.
[6] J. Awrejcewicz and C.-H. Lamarque, *Bifurcation and Chaos in Nonsmooth Mechanical Systems*. Singapore: World Scientific, 2003.
[7] Y. Tang and X. Guan, "Parameter estimation for time-delay chaotic system by particle swarm optimization," *Chaos, Solitons Fractals*, vol. 40, no. 3, pp. 1391–1398, 2009.
[8] T. Heil, I. Fischer, W. Elsässer, J. Mulet, and C. R. Mirasso, "Chaos synchronization and spontaneous symmetry-breaking in symmetrically delay-coupled semiconductor lasers," *Phys. Rev. Lett.*, vol. 86, p. 795, Jan. 2001.
[9] C. Li, X. Liao, and K.-W. Wong, "Chaotic lag synchronization of coupled time-delayed systems and its applications in secure communication," *Phys. D, Nonlinear Phenomena*, vol. 194, no. 3, pp. 187–202, 2004.
[10] F. M. Atay, *Complex Time-Delay Systems: Theory and Applications*. Berlin, Germany: Springer, 2010.
[11] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, no. 8, p. 821, 1990.
[12] U. Parlitz, L. Junge, and L. Kocarev, "Synchronization-based parameter estimation from time series," *Phys. Rev. E*, vol. 54, no. 6, p. 6253, 1996.
[13] U. Parlitz, "Estimating model parameters from time series by autosynchronization," *Phys. Rev. Lett.*, vol. 76, no. 8, p. 1232, 1996.
[14] M. Pourmahmood, S. Khanmohammadi, and G. Alizadeh, "Synchronization of two different uncertain chaotic systems with unknown parameters using a robust adaptive sliding mode controller," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, no. 7, pp. 2853–2868, 2011.
[15] L. Huang, M. Wang, and R. Feng, "Parameters identification and adaptive synchronization of chaotic systems with unknown parameters," *Phys. Lett. A*, vol. 342, no. 4, pp. 299–304, 2005.
[16] Y. Xu, W. Zhou, J. Fang, and W. Sun, "Adaptive bidirectionally coupled synchronization of chaotic systems with unknown parameters," *Nonlinear Dyn.*, vol. 66, no. 1, pp. 67–76, 2011.
[17] S. Vaidyanathan, K. Madhavan, and B. A. Idowu, "Backstepping control design for the adaptive stabilization and synchronization of the pandey jerk chaotic system with unknown parameters," *Int. J. Control Theory Appl.*, vol. 9, no. 1, pp. 299–319, 2016.
[18] W. Hu, Y. Yu, and S. Zhang, "A hybrid artificial bee colony algorithm for parameter identification of uncertain fractional-order chaotic systems," *Nonlinear Dyn.*, vol. 82, no. 3, pp. 1–16, 2015.
[19] D. E. Goldberg, "Genetic algorithm in search, optimization, and machine learning," *Mach. Learn.*, vol. 3, no. 7, pp. 2104–2116, 1989.
[20] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Boston, MA, USA: Springer, 2011, pp. 760–766.
[21] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[22] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, 2009, pp. 210–214.
[23] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, 2007.
[24] M. Ahmadi and H. Mojallali, "Chaotic invasive weed optimization algorithm with application to parameter estimation of chaotic systems," *Chaos, Solitons Fractals*, vol. 45, nos. 9–10, pp. 1108–1120, 2012.
[25] J. Lin and C. Chen, "Parameter estimation of chaotic systems by an oppositional seeker optimization algorithm," *Nonlinear Dyn.*, vol. 76, no. 1, pp. 509–517, 2014.
[26] J. Lin, "Parameter estimation for time-delay chaotic systems by hybrid biogeography-based optimization," *Nonlinear Dyn.*, vol. 77, no. 3, pp. 983–992, 2014.
[27] H. Li and H. Wu, "An oppositional wolf pack algorithm for parameter identification of the chaotic systems," *Optik-Int. J. Light Electron Opt.*, vol. 127, no. 20, pp. 9853–9864, 2016.
[28] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *Int. J. Math. Model. Numer. Optim.*, vol. 1, no. 4, pp. 330–343, 2010.
[29] M. K. Marichelvam, "An improved hybrid cuckoo search (IHCS) metaheuristics algorithm for permutation flow shop scheduling problems," *Int. J. Bio-Inspired Comput.*, vol. 4, no. 4, pp. 200–205, 2012.
[30] M. K. Marichelvam, T. Prabaharan, and X. S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan," *Appl. Soft Comput.*, vol. 19, no. 1, pp. 93–101, Jun. 2014.
[31] S. Walton, O. Hassan, K. Morgan, and M. Brown, "Modified cuckoo search: A new gradient free optimisation algorithm," *Chaos, Solitons Fractals*, vol. 44, no. 9, pp. 710–718, 2011.
[32] W. Long, X. Liang, Y. Huang, and Y. Chen, "An effective hybrid cuckoo search algorithm for constrained global optimization," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 911–926, 2014.
[33] H. Rakhshani and A. Rahati, "Snap-drift cuckoo search: A novel cuckoo search optimization algorithm," *Appl. Soft Comput.*, vol. 52, pp. 771–794, Apr. 2017.
[34] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[35] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.
[36] Z. Zhang and Y. Chen, "An improved cuckoo search algorithm with adaptive method," in *Proc. Comput. Sci. Optim. (CSO)*, 2014, pp. 204–207.
[37] L. Xiang-Tao and Y. Ming-Hao, "Parameter estimation for chaotic systems using the cuckoo search algorithm with an orthogonal learning method," *Chin. Phys. B*, vol. 21, no. 5, p. 050507, 2012.
[38] Z. Sheng, J. Wang, S. Zhou, and B. Zhou, "Parameter estimation for chaotic systems using a hybrid adaptive cuckoo search with simulated annealing algorithm," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 24, no. 1, p. 013133, 2014.
[39] G.-G. Wang, A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "A new hybrid method based on krill herd and cuckoo search for global optimisation tasks," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 5, pp. 286–299, 2016.
[40] X.-S. Yang, "Cuckoo search and firefly algorithm," in *Studies in Computational Intelligence*, vol. 516. Cham, Switzerland: Springer, 2014.
[41] N. H. Awad, M. Z. Ali, P. N. Suganthan, and E. Jaser, "A decremental stochastic fractal differential evolution for global numerical optimization," *Inf. Sci.*, vol. 372, pp. 470–491, Dec. 2016.
[42] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. New York, NY, USA: Springer, 2006.
[43] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Comput. Intell. Modelling, Control Autom.*, vol. 1. Nov. 2005, pp. 695–701.
[44] H. Xiao and Y. Duan, "Cuckoo search algorithm based on differential evolution," *J. Comput. Appl.*, vol. 34, no. 6, p. 1631, 2014.
[45] X. Li and M. Yin, *Modified Cuckoo Search Algorithm With Self Adaptive Parameter Method*. Amsterdam, The Netherlands: Elsevier, 2015.
[46] P. Zhao and H. Li, "Opposition-based cuckoo search algorithm for optimization problems," in *Proc. 5th Int. Symp. Comput. Intell. Des.*, 2013, pp. 344–347.
[47] H. Wang, W. Wang, H. Sun, Z. Cui, S. Rahnamayan, and S. Zeng, "A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems," *Soft Comput.*, vol. 21, no. 15, pp. 4297–4307, 2017.
[48] M. K. Naik and R. Panda, "A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition," *Appl. Soft Comput.*, vol. 38, pp. 661–675, Jan. 2016.
[49] L. Wang, Y. Zhong, and Y. Yin, "Nearest neighbour cuckoo search algorithm with probabilistic mutation," *Appl. Soft Comput.*, vol. 49, pp. 498–509, Dec. 2016.
[50] Y. Tang, X. Zhang, C. Hua, L. Li, and Y. Yang, "Parameter identification of commensurate fractional-order chaotic system via differential evolution," *Phys. Lett. A*, vol. 376, no. 4, pp. 457–464, 2012.
[51] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, no. 2, pp. 130–141, 1963.
[52] O. E. Rössler, "An equation for continuous chaos," *Phys. Lett. A*, vol. 57, no. 5, pp. 397–398, 1976.
[53] A. Namajūnas, K. Pyragas, and A. Tamaševičius, "An electronic analog of the mackey-glass system," *Phys. Lett. A*, vol. 201, no. 1, pp. 42–46, 1995.

[54] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

JIAMIN WEI received the B.S. degree from the Department of Mathematical Science, Beijing Jiaotong University, China, in 2014. She is currently pursuing the master's and Ph.D. degrees. Her research interests include evolutionary computation, global optimization, chaos control and synchronization, and their applications.

YONGGUANG YU received the M.S. degree from the Department of Mathematical Science, Inner Mongolia University, China, in 2001, and the Ph.D. degree from the Institute of Applied Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, China, in 2004. From 2007 to 2009, he was a Research Fellow with the City University of Hong Kong, China. Since 2010, he has been a Professor with the Department of Mathematics, School of Science, Beijing Jiaotong University, China. His research interests include intelligent computation, chaos control and synchronization, complex networks, and nonlinear control and multi-agent systems.