Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 2959370, 8 pages
http://dx.doi.org/10.1155/2016/2959370

Research Article

An Improved Cuckoo Search Optimization Algorithm for the Problem of Chaotic Systems Parameter Estimation

Jun Wang,[1] Bihua Zhou,[1] and Shudao Zhou[2]

[1] *National Key Laboratory on Electromagnetic Environmental Effects and Electro-Optical Engineering, PLA University of Science and Technology, Nanjing 210007, China*
[2] *College of Meteorology and Oceanography, PLA University of Science and Technology, Nanjing 211101, China*

Correspondence should be addressed to Jun Wang; wangjun19007@163.com

Received 7 June 2015; Revised 27 August 2015; Accepted 1 October 2015

Academic Editor: Jens Christian Claussen

This paper proposes an improved cuckoo search (ICS) algorithm to establish the parameters of chaotic systems. In order to improve the optimization capability of the basic cuckoo search (CS) algorithm, the orthogonal design and simulated annealing operation are incorporated in the CS algorithm to enhance the exploitation search ability. Then the proposed algorithm is used to establish parameters of the Lorenz chaotic system and Chen chaotic system under the noiseless and noise condition, respectively. The numerical results demonstrate that the algorithm can estimate parameters with high accuracy and reliability. Finally, the results are compared with the CS algorithm, genetic algorithm, and particle swarm optimization algorithm, and the compared results demonstrate the method is energy-efficient and superior.

1. Introduction

Chaos is a universal complex dynamical phenomenon, lurking in many nonlinear systems, such as communication systems and meteorological systems. The control and synchronization of chaos has been widely studied [1–4]. Parameter estimation is a prerequisite to accomplish the control and synchronization of chaos. During recent years many parameter estimation methods have been proposed, such as particle swarm optimization (PSO) [5–8], genetic algorithm (GA) [9–12], and mathematical methods of multiple shooting [13]. However, the GA and PSO algorithms are easily trapped into local-best solution that affects the quality of solutions; the precisions of PSO, GA, and multiple shooting are not high enough. Recently, a novel and robust metaheuristic based method called cuckoo search algorithm was proposed by Yang and Deb [14–16]. The algorithm proved to be very promising and could outperform existing algorithms such as GA and PSO [14]. However, the relatively poor ability of local searching is a drawback, and it is necessary to further improve the performance of CS algorithm to obtain a higher-quality solution. The basic principle of the ICS algorithm is to integrate the orthogonal design and simulated annealing operation to enhance the exploitation optimization capacity.

The remaining sections of this paper are organized as follows. In Section 2, a brief formulation of chaotic system parameters estimation is described. Section 3 elaborates the ICS algorithm, and the results established upon the proposed algorithm and some compared algorithms are given in Section 4. The paper ends with conclusions in Section 5.

2. Problem Formulation

A problem of parameter estimation can be converted into a problem of multidimensional optimization by constructing the proper fitness function.

Let the following equation be a continuous nonlinear n-dimension chaotic system:

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

where $X = (x_1, x_2, \ldots, x_n)^T \in R^n$ denotes the state vector of the chaotic system, $\dot{X}$ is the derivative of X, $X_0 = (x_{10}, x_{20}, \ldots, x_{n0})^T \in R^n$ denotes the initial state of system, and

```
Begin
  Generate iteration time $t = 1$
  Initialized with random vector values and parameters
  Evaluate the fitness of each individual (nest) and determine the best individual with the best fitness value
  While (stopping criterion is not met or $t <$ MaxGeneration)
    Get a Cuckoo randomly by local random walk or Lévy Flights
    Evaluate its fitness $F_m$
    Choose a nest among $n$ (say, $n$) randomly
    If ($F_m < F_n$)
      replace $n$ by the new nest $m$
    End If
    A fraction ($p_a$) of worse nests are abandoned and new ones are built
    Keep the best nests with quality solutions
    Rank the solutions and find the current best
    Update the generation number $t = t + 1$
  End while
End
```

Algorithm 1: Basic cuckoo search algorithm.

$\theta = (\theta_1, \theta_2, \ldots, \theta_d)^T$ is a set of original parameters. Suppose the structure of the system (1) is known; then the estimated system can be written as

$$\dot{\widetilde{X}} = F\left(\widetilde{X}, X_0, \widetilde{\theta}\right), \qquad (2)$$

where $\widetilde{X} = (\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_n)^T \in R^n$ denotes the state vector of the estimated system; $\widetilde{\theta} = (\widetilde{\theta}_1, \widetilde{\theta}_2, \ldots, \widetilde{\theta}_d)^T$ is a set of estimated parameters. In order to convert the parameter estimation problem into optimization problem, the following objective fitness function is defined:

$$F\left(\widetilde{\theta}\right) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(X - \widetilde{X}\right)^2}, \qquad (3)$$

where $i = 1, 2, \ldots, M$ is the sampling time point and M denotes the length of data used for parameter estimation. The parameter estimation of system (1) can be achieved by searching the most proper values of $\widetilde{\theta}$ such that the objective function (3) is globally minimized.

It can be found that (3) is a multidimensional nonlinear function with multiple local search optima; it is easily trapped into local optimal solution and the computation amount is great, so it is not easy to search the globally optimal solution effectively and accurately using traditional general methods. In the paper an improved CS algorithm is proposed to solve the complex optimization problem.

3. Improved CS Algorithm

3.1. Basic CS Algorithm. The basic CS algorithm is based on the brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds. For simplicity in describing the basic CS, the following three ideal rules are used [14]: (1) Each cuckoo lays one egg at a time, and dumps it in a randomly chosen set; (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 egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. In this case, the host bird can either get rid of the egg away or simply abandon the nest and build a complex new nest. Based on the above rules, the basic CS algorithm is described as shown in Algorithm 1 [14].

Furthermore, the algorithm used a balanced combination of a local random walk and the global explorative random walk, controlled by a switching parameter p_a. The local random walk can be written as

$$x_i^{t+1} = x_i^t + \alpha s \otimes H\left(p_a - \varepsilon\right) \otimes \left(x_j^t - x_k^t\right), \qquad (4)$$

where x_j^t and x_k^t are two different solutions selected randomly by random permutation, H is a Heaviside function, ε is a random number drawn from a uniform distribution, and s is the step size.

On the other hand, the global random walk is carried out by using Lévy flights [14–17]:

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}\,(s, \lambda). \qquad (5)$$

Here, $\alpha > 0$ is the step size scaling factor; Lévy(s, λ) is the step-lengths that are distributed according to the following probability distribution shown in (6) which has an infinite variance with an infinite mean:

$$\text{Lévy}\,(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}. \qquad (6)$$

3.2. ICS Algorithm. In order to further improve searching ability of the algorithm, the orthogonal design and simulated annealing operation are integrated into the CS algorithm. The basic idea of the orthogonal design is to utilize the properties of the fractional experiment to efficiently determine the best combination of levels [17]. An orthogonal array of K factors with Q levels and M combinations is denoted as $L_M(Q^K)$,

```
Step 1. Construct the basic columns
  For $k = 1$ to $J$
    $j = \frac{Q^{k-1} - 1}{Q - 1} + 1$
    For $i = 1$ to $Q^J$
      $a_{i,j} = \left\lfloor \frac{i-1}{Q^{J-K}} \right\rfloor \bmod Q$
    End for
  End for
Step 2. Construct the non-basic columns
  For $k = 2$ to $J$
    $j = \frac{Q^{k-1} - 1}{Q - 1} + 1$
    For $s = 1$ to $j - 1$
      For $t = 1$ to $Q - 1$
  $a_{j+(s-1)(Q-1)+1} = \left(a_s \times t + a_j\right) \bmod Q$
      End for
    End for
  End for
Step 3. Increment $a_{i,j}$ by one for $1 \le i \le M, 1 \le j \le N$
```

Procedure 1: Procedure constructing the orthogonal array.

```
Begin
(1) Construct the orthogonal array following the above steps
(2) Randomly select two solutions from the population
(3) Quantize the domain formed by the two solutions
(4) Randomly generate $k - 1$ integers $p_1 \cdots p_{k-1}$
(5) Use $L_M(Q^K)$ to generate $M$ potential offspring
(6) Randomly select a solution from the population
(7) Compare the solution with the best solution from the orthogonal $M$ offspring and obtain the better solution
End
```

Algorithm 2: Orthogonal design algorithm.

```
Begin
(1) Given a configuration of the elements of the system, randomly displace the elements on a time,
    by a small amount and calculate the resulting change in the energy, $\Delta E$
(2) If $\Delta E < 0$
    Then accept the displacement and use the resulting configuration as the starting point for the next iteration
    Else
    Accept the displacement with probability $P(\Delta E) = \exp(-\Delta E / k_b T)$ where $T$ is the current temperature and
    $k_b$ is Boltzmann's constant
(3) Repetition of this step continues until equilibrium is achieved
End
```

Algorithm 3: Simulated annealing algorithm.

where Q is the prime number, $M = Q^J$, and J is a positive integer satisfying $K = (Q^J - 1)/(Q - 1)$. The brief procedure of constructing the orthogonal array $L_M(Q^K) = [a_{i,j}]_{M \times K}$ is described as shown in Procedure 1.

The procedure of the orthogonal design algorithm is elaborated as shown in Algorithm 2 and for more detailed information on the orthogonal design strategy, please refer to [17–19].

The procedure of simulated annealing algorithm is simply stated as shown in Algorithm 3 [20], and for more detailed information on the simulated annealing, please refer to [20–22].

```
Begin
(1) Initial the parameter values of the algorithm, generate the random initial vector values and set the iteration number $t = 1$.
(2) Evaluate fitness values of each individual and determine the current best individual with the best objective value.
    Check whether the stopping criterion is met. If the stopping criterion is met, then output the best solution;
    otherwise update the iteration number $t = t + 1$ and continue the iteration process until the stopping criterion is met.
(3) Keep the best solution of the last iteration, and get a set of new solutions $X_{\text{new}} = [x_1{}^{(t+1)}, \ldots, x_i{}^{(t+1)}, \ldots, x_K{}^{(t+1)}]$ by
    Lévy flight, the Lévy flight is performed according to (4).
(4) Evaluate the fitness value $F_i{}^{(t+1)}$ of the new solution $x_i{}^{(t+1)}$, and compare $F_i{}^{(t+1)}$ with $F_i{}^{(t)}$ which represents the solution of
    the $t$th iteration. If $F_i{}^{(t+1)}$ is better than $F_i{}^{(t)}$, then replace $x_i{}^{(t)}$ by $x_i{}^{(t+1)}$, otherwise, not abandoning the solution $x_i{}^{(t+1)}$
    at once but accepting the solution with probability $p = e^{(-|\Delta f|/(K \cdot T))}$, where $\Delta f$ is the change in the fitness value
    $\Delta f = F_i{}^{(t+1)} - F_i{}^{(t)}$. $K$ is Boltzmann's constant. $T$ is the current temperature. Select a random variable $r_i \in [0, 1]$, $i \in [1, N]$,
    if $p \geq r_i$, then accept the new solution and use the solution as the starting point for the next iteration. Otherwise,
    abandon the solution, then a set of solution $X'_{\text{new}} = [x_{11}{}^{(t+1)}, x_{22}{}^{(t+1)}, \ldots, x_{ii}{}^{(t+1)}]$ are obtained.
(5) A fraction ($p_a$) of worse nests are abandoned and new ones are built.
(6) Implement the orthogonal design strategy procedures.
(7) Go to step (2)
End
```

Algorithm 4: Improved cuckoo search algorithm.

Based on the above description of the orthogonal design strategy and simulated annealing operation, the detailed procedures for parameter estimation with the ICS algorithm can be summarized as shown in Algorithm 4.

4. Simulation Results

To demonstrate the effectiveness of the improved algorithm, the algorithm is used to estimate parameters of Lorenz chaotic system [23] and Chen chaotic system [24].

4.1. Lorenz Chaotic System. Lorenz chaotic system equation [23] is expressed as follows:

$$\begin{aligned} \dot{x} &= \sigma_1 (y - x), \\ \dot{y} &= \sigma_2 x - xz - y, \\ \dot{z} &= xy - \sigma_3 z, \end{aligned} \tag{7}$$

where (x, y, z) is the state variables; $\sigma_1, \sigma_2, \sigma_3$ are the unknown chaotic system parameters which need to be estimated. The real parameters of the system are $\sigma_1 = 10$, $\sigma_2 = 28$, and $\sigma_3 = 8/3$ which ensure a chaotic behavior, in order to obtain the values of some state variables, the fourth-order Runge-Kutta algorithm is used to solve (7), and the integral step is $h = 0.01$. Then a series of state variables values are obtained and 100 state variables of different times $(\{(x(n), y(n), z(n)),\ n = 1, 2, \ldots, 100\})$ are chosen to be the sample data. The parameters of the algorithm are set as follows: the max iteration number is $N = 200$, the sample size is $M = 100$, the annealing mode is shown in (8) where n is the iteration number, and the initial temperature is $T_0 = 100$. Consider

$$T(n) = \frac{T_0}{\ln(1 + n)}, \tag{8}$$

The objective (fitness) function H is shown in (9), where $(x(n), y(n), z(n))$ is the nth state variable that corresponds to the true system parameters and $(\tilde{x}(n), \tilde{y}(n), \tilde{z}(n))$ is the nth state variable that corresponds to the estimated system parameters:

$$H = \sqrt{\frac{1}{M} \sum_{n=1}^{M} \left[(\tilde{x}(n) - x(n))^2 + (\tilde{y}(n) - y(n))^2 + (\tilde{z}(n) - z(n))^2 \right]}. \tag{9}$$

Figure 1 shows the convergence process of the fitness values and three parameters $(\sigma_1, \sigma_2, \sigma_3)$ during the iterations in a single experiment.

In order to eliminate the difference of each experiment, the algorithm is also executed 50 times; then the mean value of the 50 experiments is taken as the final estimated value; the mean value and best value of the 50 experiments are listed in Table 1. The results based on CS (the best parameter setting is $p_a = 0.25$, $p_a = 0.01$), PSO (the best parameter setting is $w = 0.8$, $c = 1.5$, where w is the inertia weight and c is acceleration factor), and GA (the best parameter setting is $c_r = 0.8$, $m_u = 0.1$, where c_r is the crossover rate and m_u is the mutation rate) are also listed in Table 1.

It can be seen from Table 1 that the best fitness values obtained by ICS algorithm are quite better than the other algorithms. The mean values of the established parameters are also with higher precision than others. The estimated values are close to the true values infinitely. It can be concluded in general that the ICS algorithm contributes to superior performance, CS performs nest-best, PSO is better than GA, and GA performs worst.

As the actual chaotic systems always associate with noise, in order to test the performance of parameter estimation in the noise condition, the noise sequences are added to the original sample data. The white noise is added to the state variables $\{(x(n), y(n), z(n)),\ n = 1, 2, \ldots, 100\}$; the range of the noise sequences is from -0.1 to 0.1. Figure 2 shows the convergence process of the fitness values and

Table 1: The statistical results based on different methods in the noiseless condition.

	Mean value				Best value			
	ICS	CS	PSO	GA	ICS	CS	PSO	GA
σ_1	10.000000	9.998736	9.985012	10.082051	10.000000	9.999927	9.995510	10.026911
σ_2	28.000000	28.000005	28.014411	27.881034	28.000000	28.000002	28.001304	28.004702
σ_3	2.666667	2.666661	2.668102	2.681882	2.666667	2.666665	2.666802	2.669018
H	$1.1822e - 010$	$3.7614e - 004$	0.069517	0.331901	$1.2933e - 011$	$2.9556e - 005$	0.011377	0.113969

Table 2: The statistical results by different algorithms in the noise condition.

	Mean value				Best value			
	ICS	CS	PSO	GA	ICS	CS	PSO	GA
σ_1	9.996110	10.080014	9.844606	10.217998	9.998941	10.001565	9.881002	10.044011
σ_2	28.002272	27.980212	27.860013	27.661201	28.000099	28.001995	28.022441	27.900189
σ_3	2.666590	2.658890	2.700198	2.659880	2.666675	2.667704	2.675596	2.670228
H	0.008402	0.0390389	0.221401	0.500227	0.000908	0.001228	0.050931	0.255996

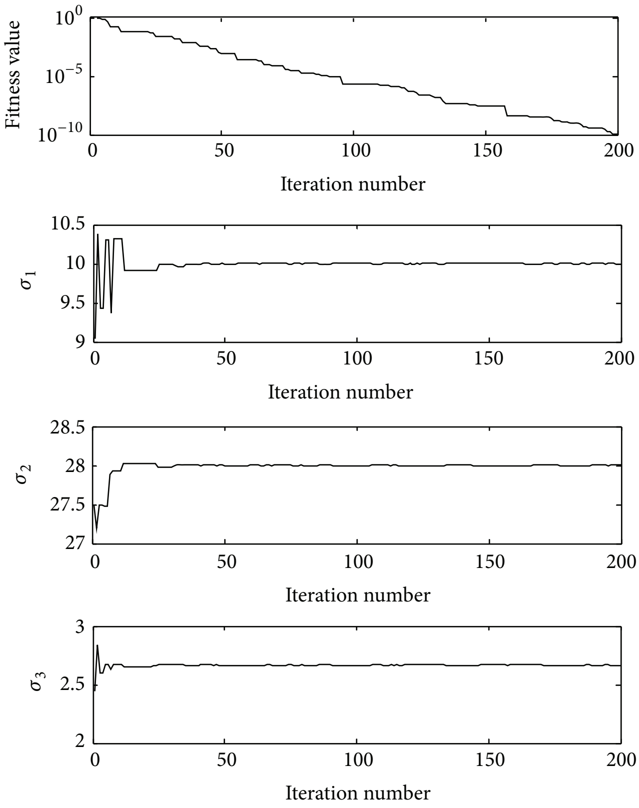

Figure 1: The convergence process of fitness function value and three parameters ($\sigma_1, \sigma_2, \sigma_3$) during the iterations in a single experiment under the noiseless condition.

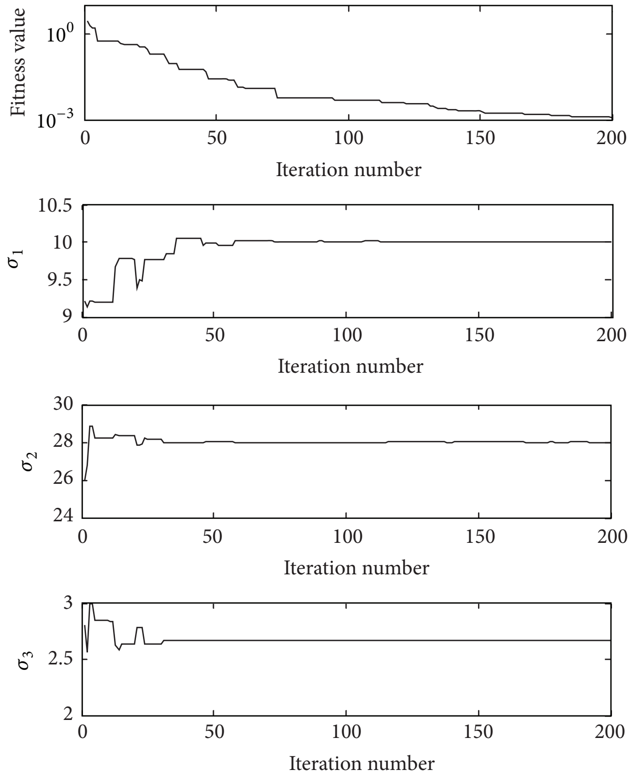

Figure 2: The convergence process of fitness function value and three parameters ($\sigma_1, \sigma_2, \sigma_3$) during the iterations in a single experiment under the noise condition.

three parameters ($\sigma_1, \sigma_2, \sigma_3$) during the iterations in a single experiment under the noise condition.

In order to eliminate the difference of each experiment, the algorithm is executed 50 times, then the mean value of the 50 experiments is taken as the final estimated value, and the corresponding results are listed in Table 2.

It can be seen from Table 2 that the four algorithms all have a certain capability of identification of parameters, but the performance of ICS is much better than the other algorithms; it supplies more robust and precise results; although the precision of the estimated parameters is declined compared with the results in the noiseless condition, the precision is still satisfactory. Then it can be concluded that the ICS

Table 3: The statistical results by different algorithms in the noiseless condition.

	Mean value				Best value			
	ICS	CS	PSO	GA	ICS	CS	PSO	GA
σ_1	34.999438	35.089675	34.844278	33.535396	34.999945	34.996661	34.782290	35.102699
σ_2	2.999951	2.999081	3.012977	3.005031	2.999999	2.999907	2.998694	2.991955
σ_3	27.999757	28.043810	27.917648	27.291109	27.999974	27.998427	27.895888	28.053975
H	$4.6438e-007$	$4.6628e-004$	0.027312	0.115259	$2.9658e-010$	$3.2051e-006$	0.003312	0.010562

Table 4: The statistical results by different algorithms in the noise condition.

	Mean value				Best value			
	ICS	CS	PSO	GA	ICS	CS	PSO	GA
σ_1	35.421272	35.859540	34.108398	35.962547	34.970096	35.112205	35.313896	10.044011
σ_2	2.996796	2.993281	2.970277	3.052984	2.999311	2.997309	2.980455	27.900189
σ_3	28.204815	28.418425	27.588450	28.434651	27.985940	28.055617	28.162608	2.670228
H	0.009334	0.038112	0.244276	0.591957	$1.5986e-004$	0.001495	0.062901	0.255996

algorithm possesses a powerful capability for parameters identification in the noise condition.

4.2. Chen Chaotic System. Chen chaotic system equation [24] is expressed as follows:

$$\begin{aligned} \dot{x} &= \sigma_1 (y - x), \\ \dot{y} &= (\sigma_3 - \sigma_1) x - \sigma_3 y - xz, \\ \dot{z} &= xy - \sigma_2 z, \end{aligned} \tag{10}$$

where (x, y, z) is the state variables; $\sigma_1, \sigma_2, \sigma_3$ are the unknown chaotic system parameters which need to be estimated. The real parameters of the system are $\sigma_1 = 35$, $\sigma_2 = 3$, and $\sigma_3 = 28$ which ensure a chaotic behavior, the fourth-order Runge-Kutta algorithm is used to solve (10), and the integral step is $h = 0.01$. Then a series of state variables values are obtained and 100 state variables of different times $(\{(x(n), y(n), z(n)),\ n = 1, 2, \ldots, 100\})$ are chosen to be the sample data. The parameters of the algorithm are set as follows: the max iteration number is $N = 200$, the sample size is $M = 100$, the annealing mode is shown in (8) where n is the iteration number, and the initial temperature is $T_0 = 100$. The convergence process of the fitness values and three parameters $(\sigma_1, \sigma_2, \sigma_3)$ during the iterations in a single experiment is shown in Figure 3. In order to eliminate the difference of each experiment, the algorithm is executed 50 times, then the mean value of the 50 experiments is taken as the final estimated value, and the corresponding results are listed in Table 2.

It can be seen from Table 3 that the best fitness values obtained by ICS algorithm are quite better than the other algorithms. The mean values of the established parameters are also with higher precision than others. The estimated values are close to the true values asymptotically. It can be concluded in general that the ICS algorithm contributes to superior performance, CS performs nest-best, PSO is better than GA, and GA performs worst.

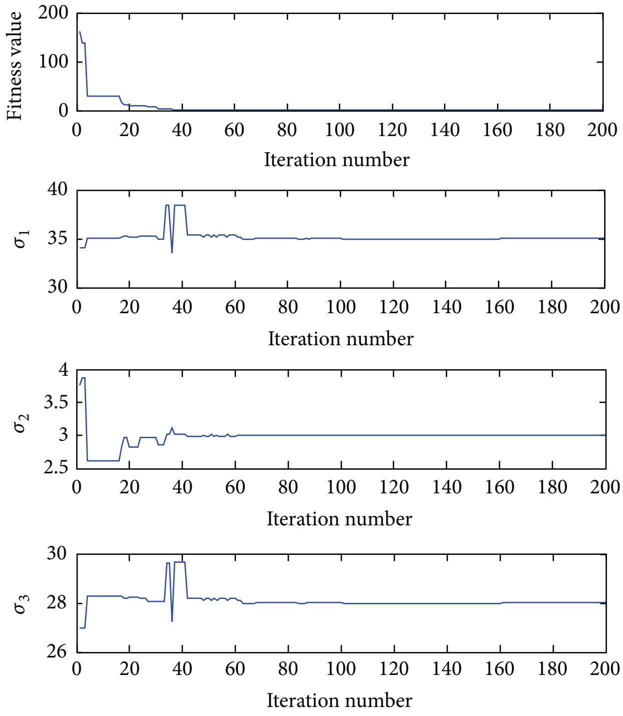

Figure 3: The convergence process of fitness function value and three parameters $(\sigma_1, \sigma_2, \sigma_3)$ during the iterations in a single experiment under the noiseless condition.

As the actual chaotic systems always come along with noise, in order to test the performance of parameter estimation in the noise condition, the noise sequences are added to the original sample data. The white noise is added to the state variables $\{(x(n), y(n), z(n)),\ n = 1, 2, \ldots, 100\}$; the range of the noise sequences is from −0.1 to 0.1. Figure 4 shows the convergence process of the fitness values and three parameters $(\sigma_1, \sigma_2, \sigma_3)$ during the iterations in a single experiment under the noise condition.

It can be seen from Table 4 that the four algorithms all have a certain capability of identification of parameters, but the performance of ICS is much better than the other algorithms; it supplies more robust and precise results; although

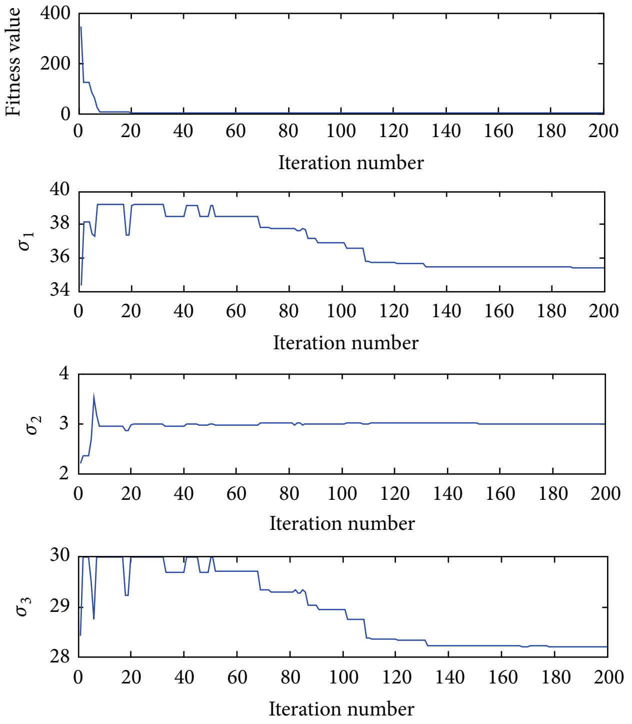

Figure 4: The convergence process of fitness function value and three parameters ($\sigma_1, \sigma_2, \sigma_3$) during the iterations in a single experiment under the noise condition.

the precision of the estimated parameters is declined compared with the results in the noiseless condition, the precision is still satisfactory. Then it can be concluded that the ICS algorithm possesses a powerful capability for parameters identification in the noise condition.

5. Conclusion

In this paper, an energy-efficient and superior ICS algorithm is proposed to estimate chaotic system parameters. The estimated results demonstrate the strong capabilities and effectiveness of the proposed algorithm, compared with the CS, PSO, and GA algorithms; the ICS algorithm supplies more robust and precise results. Besides, the algorithm also has a more powerful capability of noise immunity. In general, the proposed ICS algorithm is a feasible, energy-efficient, and promising method for parameters estimation of chaotic systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by National Natural Science Foundation of China, 61271106.

References

[1] A. Sitz, U. Schwarz, J. Kurths, and H. U. Voss, "Estimation of parameters and unobserved components for nonlinear systems from noisy time series," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 66, no. 1, Article ID 016210, 2002.

[2] L. X. Guo, M. F. Hu, Z. Y. Xu, and A. Hu, "Synchronization and chaos control by quorum sensing mechanism," *Nonlinear Dynamics*, vol. 73, no. 3, pp. 1253–1269, 2013.

[3] U. Parlitz, "Estimating model parameters from time series by autosynchronization," *Physical Review Letters*, vol. 76, no. 8, pp. 1232–1244, 1996.

[4] H. N. Agiza and M. T. Yassen, "Synchronization of Rossler and Chen chaotic dynamical systems using active control," *Physics Letters A*, vol. 278, no. 4, pp. 191–197, 2001.

[5] F. Gao and H.-Q. Tong, "Parameter estimation for chaotic system based on particle swarm optimization," *Acta Physica Sinica*, vol. 55, no. 2, pp. 577–582, 2006.

[6] Q. He, L. Wang, and B. Liu, "Parameter estimation for chaotic systems by particle swarm optimization," *Chaos, Solitons and Fractals*, vol. 34, no. 2, pp. 654–661, 2007.

[7] J. Garcia-Nieto, A. C. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 823–839, 2013.

[8] M. Salahi, A. Jamalian, and A. Taati, "Global minimization of multi-funnel functions using particle swarm optimization," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2101–2106, 2013.

[9] C. Tao, Y. Zhang, and J. J. Jiang, "Estimating system parameters from chaotic time series with synchronization optimized by a genetic algorithm," *Physical Review E*, vol. 76, no. 1, Article ID 016209, 2007.

[10] J. Wang, Z. Sheng, B. H. Zhou, and S. D. Zhou, "Lightning potential forecast over Nanjing with denoised sounding derived indices based on SSA and CS-BP neural network," *Atmospheric Research*, vol. 137, pp. 245–256, 2014.

[11] J. Wang, B. H. Zhou, S. D. Zhou, and Z. Sheng, "Forecasting nonlinear chaotic time series with function expression method based on an improved genetic-simulated annealing algorithm," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 341031, 10 pages, 2015.

[12] G. G. Szpiro, "Forecasting chaotic time series with genetic algorithms," *Physical Review E*, vol. 55, no. 3, Article ID 2557, 1997.

[13] M. Peifer and J. Timmer, "Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting," *IET Systems Biology*, vol. 1, no. 2, pp. 78–88, 2007.

[14] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09)*, pp. 210–214, IEEE, Coimbatore, India, December 2009.

[15] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.

[16] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[17] C. R. Hick, *Fundamental Concepts in the Design of Experiments*, Saunders College Publishing, Austin, Tex, USA, 1993.

[18] D. C. Montgomery, *Design and Analysis of Experiments*, Wiley, New York, NY, USA, 1991.

[19] Y.-W. Leung, Y. Wang, Y.-W. Leung, and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 41–53, 2001.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] M. S. Kim, M. R. Feldman, and C. C. Guest, "Optimum encoding of binary phase-only filters with a simulated annealing algorithm," *Optics Letters*, vol. 14, no. 11, pp. 545–547, 1989.

[22] B. L. Golden and C. C. Skiscim, "Using simulated annealing to solve routing and location problems," *Naval Research Logistics Quarterly*, vol. 33, no. 2, pp. 261–279, 1986.

[23] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[24] J. Lü and G. Chen, "A new chaotic attractor coined," *International Journal of Bifurcation and Chaos*, vol. 12, no. 3, pp. 659–661, 2002.