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

Copyright © 2016 Jun Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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:

$$X = F (X, X_0, \theta),$$

where $X = (x_1, x_2, \ldots, x_n)^T \in R^n$ denotes the state vector of the chaotic system, $X$ 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...
\[ \theta = (\theta_1, \theta_2, \ldots, \theta_d)^T \] is a set of original parameters. Suppose the structure of the system (1) is known; then the estimated system can be written as
\[ \dot{\tilde{X}} = F \left( \tilde{X}, X_0, \tilde{\theta} \right), \tag{2} \]
where \( \tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)^T \in \mathbb{R}^n \) denotes the state vector of the estimated system; \( \tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_d)^T \) is a set of estimated parameters. In order to convert the parameter estimation problem into optimization problem, the following objective function is defined:
\[ F(\bar{\theta}) = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (X - \bar{X})^2}, \tag{3} \]
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 \( \bar{\theta} \) 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
\[ x_{i,t+1}^j = x_{i,t}^j + \alpha s \odot H(p_a - \varepsilon) \odot (x_{i,t}^j - x_{k,t}^j), \tag{4} \]
where \( x_{i,t}^j \) and \( x_{k,t}^j \) are two different solutions selected randomly by random permutation, \( H \) is a Heaviside function, \( \varepsilon \) 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]:
\[ x_{i,t+1}^{j'} = x_{i,t}^{j'} + \alpha \odot \text{Lévy}(s, \lambda). \tag{5} \]
Here, \( \alpha > 0 \) is the step size scaling factor; \( \text{Lévy}(s, \lambda) \) 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:
\[ \text{Lévy}(s, \lambda) = \frac{\lambda^\lambda (\lambda/2) \sin(\pi \lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}. \tag{6} \]
**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 \( Q - 1 \)

\[ a_{j+(s-1)(Q-1)+1} = (a_s \times t + a_j) \bmod Q \]

End for

End for

End for

**Step 3. Increment \( a_{i,j} \) by one for \( 1 \leq i \leq M, 1 \leq j \leq N \)**

---

**Procedure 1: Procedure constructing the orthogonal array.**

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

**Algorithm 2: Orthogonal design algorithm.**

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

**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].
The range of the noise sequences is from -0.1 to 0.1. Figure 2 shows the convergence process of the fitness values and the iteration number, and the initial temperature is $T_M = 100$ is performed according to (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 $X_{new}^{(t+1)} = [x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_n^{(t+1)}]$ are obtained.

A fraction ($p_a$) of worse nests are abandoned and new ones are built.

Implement the orthogonal design strategy procedures.

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:

\[
\begin{align*}
\dot{x} &= \sigma_1 (y - x), \\
\dot{y} &= \sigma_2 x - xz - y, \\
\dot{z} &= xy - \sigma_3 z,
\end{align*}
\]

(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

\[
T(n) = \frac{T_0}{\ln(1 + n)}, \quad n = 1, 2, \ldots, 100.
\]

The objective (fitness) function $H$ is shown in (9), where $(x(n), y(n), z(n))$ is the $n$th state variable that corresponds to the true system parameters and $(\overline{x}(n), \overline{y}(n), \overline{z}(n))$ is the $n$th state variable that corresponds to the established parameters.

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

(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 $c_r = 0.8, m_a = 0.1$, where $c_r$ is the crossover rate and $m_a$ 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.

<table>
<thead>
<tr>
<th></th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_3$</td>
<td>2.666667</td>
<td>2.666661</td>
<td>2.668102</td>
<td>2.681882</td>
<td>2.666667</td>
<td>2.666665</td>
<td>2.666802</td>
<td>2.669018</td>
</tr>
<tr>
<td>$H$</td>
<td>1.1822e−010</td>
<td>3.764e−004</td>
<td>0.069517</td>
<td>0.331901</td>
<td>1.2933e−011</td>
<td>2.9556e−005</td>
<td>0.011377</td>
<td>0.113969</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_2$</td>
<td>28.002272</td>
<td>27.980212</td>
<td>27.860013</td>
<td>27.661201</td>
<td>28.000099</td>
<td>28.001995</td>
<td>28.022441</td>
<td>27.900189</td>
</tr>
<tr>
<td>$\sigma_3$</td>
<td>2.666590</td>
<td>2.658890</td>
<td>2.700198</td>
<td>2.659880</td>
<td>2.666675</td>
<td>2.667704</td>
<td>2.675596</td>
<td>2.670228</td>
</tr>
<tr>
<td>$H$</td>
<td>0.008402</td>
<td>0.0390389</td>
<td>0.221401</td>
<td>0.500227</td>
<td>0.000908</td>
<td>0.001228</td>
<td>0.050931</td>
<td>0.255996</td>
</tr>
</tbody>
</table>

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.

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
The statistical results by different algorithms in the noiseless condition.

<table>
<thead>
<tr>
<th></th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>34.999438</td>
<td>35.089675</td>
<td>34.844278</td>
<td>33.53396</td>
<td>34.99945</td>
<td>34.996661</td>
<td>34.782290</td>
<td>35.102699</td>
</tr>
<tr>
<td>Best</td>
<td>4.6438e−007</td>
<td>4.6628e−004</td>
<td>0.027312</td>
<td>0.115259</td>
<td>2.9658e−010</td>
<td>3.2051e−006</td>
<td>0.003312</td>
<td>0.010562</td>
</tr>
</tbody>
</table>

The statistical results by different algorithms in the noise condition.

<table>
<thead>
<tr>
<th></th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
<th>ICS</th>
<th>CS</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>35.421272</td>
<td>35.859540</td>
<td>34.108398</td>
<td>35.962547</td>
<td>35.970996</td>
<td>35.112205</td>
<td>35.313896</td>
<td>10.044011</td>
</tr>
<tr>
<td>Best</td>
<td>0.009334</td>
<td>0.038112</td>
<td>0.591957</td>
<td>1.5986</td>
<td>0.001495</td>
<td>0.010562</td>
<td>0.003312</td>
<td>0.255996</td>
</tr>
</tbody>
</table>

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:

\[
\begin{align*}
\dot{x} &= \sigma_1(y - x), \\
\dot{y} &= (\sigma_2 - \sigma_3) x - \sigma_1 y - xz, \\
\dot{z} &= xy - \sigma_2 z,
\end{align*}
\]

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 \(\sigma_1 = 35, \sigma_2 = 3, \sigma_3 = 28\) 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.

![Figure 3: The convergence process of fitness function value and three parameters during the iterations in a single experiment under the noiseless condition.](image)

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.

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...
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


