Chaotic invasive weed optimization algorithm with application to parameter estimation of chaotic systems

Mohamadreza Ahmadi, Hamed Mojallali *

Electrical Engineering Department, Faculty of Engineering, University of Guilan, Rasht, Iran

A R T I C L E   I N F O

Article history:
Received 12 October 2010
Accepted 27 May 2012
Available online 20 July 2012

A B S T R A C T

This paper introduces a novel hybrid optimization algorithm by taking advantage of the stochastic properties of chaotic search and the invasive weed optimization (IWO) method. In order to deal with the weaknesses associated with the conventional method, the proposed chaotic invasive weed optimization (CIWO) algorithm is presented which incorporates the capabilities of chaotic search methods. The functionality of the proposed optimization algorithm is investigated through several benchmark multi-dimensional functions. Furthermore, an identification technique for chaotic systems based on the CIWO algorithm is outlined and validated by several examples. The results established upon the proposed scheme are also supplemented which demonstrate superior performance with respect to other conventional methods.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Nonlinear optimization problems arise almost in all of engineering problems. Practically, many engineering problems lack continuity of variables which is a chief necessity for exploiting derivative based optimization methods. Therefore, there has been an increasing interest in applying meta-heuristic algorithms to solve nonlinear optimization algorithms in recent years. The common feature of the nature-inspired meta-heuristic algorithms is that they compromise the principles and the stochastic properties of the natural phenomena. A considerable fraction of literature has been devoted to investigate the bio-mimicry methods aimed at solving optimization problems and designing autonomous intelligent systems [1–4]. These methodologies can be roughly divided into two major categories: the evolutionary algorithms (EAs) such as GA and the swarm intelligence (SI) algorithms like PSO, and ACO.

Following this trend, the bio-inspired IWO algorithm was introduced by Mehrabian and Lucas [5] which imitates the colonial behavior of invasive weeds in nature. The IWO algorithm has shown to be virtuous in converging to optimal solution by employing some basic characteristics of weed colonization, e.g. seeding, growth and competition. Previously, the IWO algorithm has been utilized in a myriad of applications including optimizing and tuning of a robust controller [5], antenna configuration optimization [6], optimal arrangement of piezoelectric actuators on smart structures [7], DNA computing [8], and etc.

Chaos, a universal complex dynamical phenomenon, lurks in nonlinear systems, and is characterized by its ergodicity, certainty, and regularity. Despite the appearance of chaotic sequences which follow an apparently unpredictable non-periodic stochastic pattern, they can be generated by determinate equations. The first serious discussions and analyses of chaos emerged in 1963 by Lorenz [9]. Henceforth, chaos and generally chaotic properties has been widely studied and applied by scholars in different fields of science, such as MEMs, pattern recognition, optimization theory, nonlinear circuits, and so forth [10–13]. The chaotic optimization algorithm (COA) delineated in Ref. [14] adopts chaotic sequences to distribute possible solutions. The use of chaotic sequences in lieu of random
variables is at the core of chaotic optimization. According to the numerical results given in Ref. [14], the chaotic search can escape from local optima more easily compared with other stochastic search optimization algorithms, and exposes superior hill-climbing ability.

However, COA has a number of problems in practice. Chaotic sequences are extremely sensitive on initial conditions and slow in locating the optimal area as the search space extends. Thus, appropriate initial values should be tuned carefully beforehand, and the solution space should be confined.

Debate continues about alternative optimization strategies to overcome the detriments of COA. One avenue that researchers have followed in their attempt to find more robust algorithms is through integrating meta-heuristic algorithms with COA [15–22]. In [23], authors suggested a hybrid optimization algorithm incorporating chaos and tabu search. Zilong et al. [13] combined some characteristics of simulated annealing with those of chaotic optimization. Xiang et al. [18] created an improved PSO algorithm using the piece-wise linear chaotic map bringing forward the chaotic PSO (CPSO) algorithm. Yong et al. [24] combined the GA with COA in order to be exploited in neural networks.

The objective of this paper is to propose a novel optimization method by incorporating the COA and the IWO algorithm. Different chaotic maps are applied to the algorithm, and the most computational efficient one is selected accordingly. Afterwards, the algorithm is assessed to optimize benchmark functions. It is shown that the proposed CIWO method outperforms other methods like IWO and PSO which rely only on random distribution. Furthermore, the CIWO algorithm is utilized for parametric identification of chaotic systems. In order to furnish a better insight into the capability of the CIWO algorithm, the results obtained from other methods, e.g., IWO, PSO, CPSO, GA, and COA, are also included. The corresponding results verify the CIWO and the corresponding identification scheme’s accurateness.

The balance of this paper proceeds as follows. The second section of this paper briefly reviews the traditional IWO method. Section 3 considers the proposed CIWO algorithm. The numerical results established upon the proposed scheme are given in Section 4. A parameter estimation strategy using the CIWO algorithm is proposed and tested in Section 5. The paper ends with conclusions in Section 6.

2. Invasive weed optimization

2.1. Key terms

Prior to describing the IWO algorithm, the key terms are explained as follows:

Seed: each unit in the colony which encompasses a value for each variable in the optimization problem before fitness evaluation.
Weed/plant: any seed that is evaluated grows to a weed or plant.
Fitness: a value corresponding to the goodness of each unit after being evaluated.

Field: the search/solution space.
Maximum weed population: a parameter preset representing the maximum number of possible weeds in the field.

2.2. Description of traditional IWO method

The process flow of the IWO algorithm is outlined below:

1. Randomly distribute the initial seeds $S_i = \{x_1, x_2, \ldots, x_n\}$, where $n$ is the number of selected variables, over the search space. Consequently, each seed contains random values for each variable in the $n - D$ solution space.
2. The fitness of each individual seed is calculated according to the optimization problem, and the seeds grow to weeds able to produce new units.
3. Each individual is ranked based on its fitness with respect to other weeds. Subsequently, each weed produces new seeds depending on its rank in the population. The weeds which have acquired more resources have a better chance of producing seeds, and those which are less adapted to the field are unlikely to reproduce thereby creating less seeds. That is, the number of seeds to be created by each weed alters linearly from $N_{\text{min}}$ to $N_{\text{max}}$ which can be computed using the equation given below

$$\text{Number of seeds} = \frac{F_i - F_{\text{worst}}}{F_{\text{best}} - F_{\text{worst}}} (N_{\text{max}} - N_{\text{min}}) + N_{\text{min}}$$

(1)

In which $F_i$ is the fitness of $i$th weed, $F_{\text{worst}}$, and $F_{\text{best}}$ denote the best and the worst fitness in the weed population. This step ensures that each weed take part in the reproduction process.
4. The generated seeds are normally distributed over the field with zero mean and a varying standard deviation of $\sigma_{\text{iter}}$ described by

$$\sigma_{\text{iter}} = \left(\frac{\text{iter}_{\text{max}} - \text{iter}}{\text{iter}_{\text{max}}}ight)^n (\sigma_0 - \sigma_f) + \sigma_f$$

(2)

where $\text{iter}_{\text{max}}$ and $\text{iter}$ are the maximum number of iteration cycles assigned by the user, and the current iteration number respectively. $\sigma_0$ and $\sigma_f$ represent the pre-defined initial and final standard deviations. $n$ is called the nonlinear modulation index. This is a relatively critical parameter which can influence the convergence performance of the IWO algorithm. Through a set of simulations, it has been discerned that the most appropriate value for modulation index is 3 (see Fig. 1. which portrays the decrease in normalized standard deviation for different nonlinear modulation indices as the number of iteration cycles augment). Having selected this suitable value for nonlinear modulation index, the algorithm starts with a relatively high distribution variance to guarantee a complete scan of the solution space. As the iteration number increases and
the dispersion variance dwindle, the search would be restricted to the neighborhoods around the reproducing plant which has attained an apt level of fitness thereby increasing the estimation accuracy.

5. The fitness of each seed is calculated along with their parents and the whole population is ranked. Those weeds with less fitness are eliminated through competition and only a number of weeds remain which are equal to Maximum weed population.

6. The procedure is repeated at step 2 until the maximum number of iterations allowed by the user is reached.

2.3. Advantages and disadvantages of implementing IWO

On the one hand, The IWO algorithm certifies that all possible candidates would participate in the reproduction process. In contrast, most meta-heuristic algorithms would not allow the less-fitted individuals to produce offspring such as the GA. Besides, the IWO algorithm is straightforward and it includes less deal of computational burden unlike other methods. As a good illustration, one can consider the PSO algorithm. PSO needs to update both the position and velocity of individuals in each iteration round which require some extra calculations to find the best position in the neighborhood of each particle as well as the whole population [6].

On the other hand, in case of problems with an outsized search space, one has to apply a greater number of seeds to each plant so that the search space can be completely inspected. This would augment the computation time dramatically, not to mention the fact that the problem becomes even more severe as the number of variables to
be tuned increases. This is because the search should be performed in a bulkier multi-dimensional space. In addition, the gradual reduction of standard variance which plays a key-role in the IWO method can bring about immature convergence, and sometimes it is rather difficult to make an adequate trade-off between approximation preciseness and avoiding convergence to local optima.

For a more comprehensive comparison between IWO and previously suggested meta-heuristic algorithms the interested reader is referred to Refs. [5,6].

3. The Chaotic invasive weed optimization algorithm

In order to overcome the shortcomings of IWO, chaotic search is integrated in the IWO algorithm. It is worth noting that chaotic search methods have a greater ability to escape from the local minima [14]; therefore, the CIWO algorithm has a lesser chance of pre-mature convergence compared to IWO. Besides, due to greater scanning and search capabilities, the implementation of the chaotic sequences precludes the need for increasing the number of seeds in the solution space, thereby relatively reducing the computational cost of the overall algorithm.

In this section, firstly the chaotic maps utilized in the CIWO algorithm are explained. Afterwards, the proposed CIWO scheme is described. For more information regarding chaotic dynamics refer to [25].

3.1. Chaotic maps

3.1.1. Logistic map

One of the simplest and well-known chaotic maps which has been utilized in several studies [20,21,23–26], is the logistic map introduced by Sir Robert May in 1976. An implication of this is the possibility that a simple deterministic dynamic system can expose complex chaotic behavior devoid of any stochastic disruptions. The logistic map is given by

$$x_{k+1} = ax_k(1 - x_k)$$  \hspace{1cm} (3)

where $a \in [0,4]$ is the control parameter and $x \in [0,1]$ stands for the chaotic variable. By manipulating the control...
parameter one can determine whether the system is in the chaotic state, or in the stable state. The bifurcation diagram (or Feigenbaum diagram) for logistic map, which shows the distribution of $x$ against different values of $a$, is depicted in Fig. 2. The chaotic behavior of the sequence is ensured when $a = 4$, provided that the initial value for the chaotic variable ($x$) is in the range of (0,1) except for points $x = \{0.25, 0.5, 0.75\}$.

### 3.1.2. Sinusoidal map

The sinusoidal map or the sinusoid iterator is defined by

$$x_{k+1} = ax_k^2 \sin(\pi x_k)$$

which ensures chaotic behavior in the span of (0,1). Fig. 3 shows the variations of the chaotic variable versus changes in the control parameter. As it can be seen, the performance of the system becomes chaotic when $a = 2.3$. It is apparent from Figs. 2 and 3 that the logistic map and the sinusoidal map are analogous.

### 3.1.3. Tent map

The tent map exhibits chaotic dynamics. This mapping generates chaotic sequences in data range (0,1). The following equation defines the tent map:

$$x_{k+1} = \begin{cases} ax_k & x < 0.5 \\ a(1 - x_k) & x \geq 0.5 \end{cases}$$

for $a = 2$.

### 3.2. Chaotic invasive weed optimization

The goal of the optimization algorithm is to minimize

$$f(x_1, x_2, \ldots, x_m)$$

subject to

$$x_{min} < x_i < x_{max} \quad i = 1, \ldots, m$$

Table 1

<table>
<thead>
<tr>
<th>$\Delta_{max}$</th>
<th>$\sigma_0$</th>
<th>$\sigma_f$</th>
<th>$N_{max}$</th>
<th>$N_{min}$</th>
<th>Maximum weed population</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.01</td>
<td>0.00001</td>
<td>5</td>
<td>1</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Max</th>
<th>Min</th>
<th>Median</th>
<th>SD</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIWO</td>
<td>1.0001</td>
<td>1.0004</td>
<td>0.9999</td>
<td>1.0001</td>
<td>1.6728e−4</td>
<td>1.9545e−7</td>
</tr>
<tr>
<td>Logistic map</td>
<td>1.0001</td>
<td>1.0002</td>
<td>0.9997</td>
<td>1.0000</td>
<td>2.1542e−4</td>
<td>1.9053e−7</td>
</tr>
<tr>
<td>Sinusoidal map</td>
<td>1.0565</td>
<td>1.1456</td>
<td>0.9964</td>
<td>1.0165</td>
<td>0.0711</td>
<td>0.0362</td>
</tr>
<tr>
<td>Tent map</td>
<td>0.9998</td>
<td>1.002</td>
<td>0.9966</td>
<td>1.0000</td>
<td>2.7431e−4</td>
<td>2.1116e−7</td>
</tr>
<tr>
<td>PSO</td>
<td>1.0003</td>
<td>1.0210</td>
<td>0.9752</td>
<td>1.0031</td>
<td>6.7839e−4</td>
<td>2.5033e−7</td>
</tr>
<tr>
<td>CPSO</td>
<td>1.0104</td>
<td>1.1298</td>
<td>0.9874</td>
<td>0.9939</td>
<td>0.0923</td>
<td>0.0639</td>
</tr>
<tr>
<td>IWO</td>
<td>1.0520</td>
<td>1.2337</td>
<td>0.7508</td>
<td>0.7934</td>
<td>0.1737</td>
<td>0.1141</td>
</tr>
<tr>
<td>GA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. The quadratic function with two variable and a global minimum at $(x_1, x_2, y) = (1, 1, 0)$.
Fig. 6. The Rosenbrok's function with two variables and a global minimum at \((x_1, x_2, y) = (0,0,0)\).

Table 3
Parameter values of the CIWO algorithm for solving the Rosenbrok function minimization problem.

<table>
<thead>
<tr>
<th>Itermax</th>
<th>(\sigma_0)</th>
<th>(\sigma_f)</th>
<th>(N_{\text{max}})</th>
<th>(N_{\min})</th>
<th>Maximum weed population</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.025</td>
<td>0.00001</td>
<td>5</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 4
The numerical results of Rosenbrok function optimization.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Max</th>
<th>Min</th>
<th>Median</th>
<th>SD</th>
<th>(f(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIWO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic map</td>
<td>(-4.6824\times10^{-6})</td>
<td>(3.0175\times10^{-4})</td>
<td>(-5.2036\times10^{-5})</td>
<td>(-4.5788\times10^{-5})</td>
<td>(1.8445\times10^{-4})</td>
<td>(2.702\times10^{-6})</td>
</tr>
<tr>
<td>Sinusoidal map</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tent map</td>
<td>(5.1627\times10^{-4})</td>
<td>(7.4979\times10^{-4})</td>
<td>(3.1353\times10^{-4})</td>
<td>(4.7165\times10^{-4})</td>
<td>(1.6140\times10^{-4})</td>
<td>(2.8507\times10^{-4})</td>
</tr>
<tr>
<td>PSO</td>
<td>0.0008</td>
<td>0.0003</td>
<td>0.9996</td>
<td>0.0001</td>
<td>0.0004</td>
<td>6.0127\times10^{-4}</td>
</tr>
<tr>
<td>CPSO</td>
<td>(2.8428\times10^{-4})</td>
<td>(5.2628\times10^{-4})</td>
<td>(-7.8236\times10^{-5})</td>
<td>(-2.4193\times10^{-4})</td>
<td>3.6382</td>
<td>3.4529\times10^{-6}</td>
</tr>
<tr>
<td>IWO</td>
<td>0.0089</td>
<td>0.0701</td>
<td>0.0057</td>
<td>0.0109</td>
<td>0.0842</td>
<td>0.0649</td>
</tr>
<tr>
<td>GA</td>
<td>0.0386</td>
<td>0.1592</td>
<td>0.0031</td>
<td>0.0623</td>
<td>0.1205</td>
<td>0.1641</td>
</tr>
</tbody>
</table>

Fig. 7. The Rastrigin's function with two variables and a global minimum at \((x_1, x_2, y) = (0,0,0)\).
The simulation results of Rastrigin function minimization.

Table 5

<table>
<thead>
<tr>
<th>Iteration</th>
<th>( \sigma_0 )</th>
<th>( \sigma_f )</th>
<th>( N_{\text{max}} )</th>
<th>( N_{\text{min}} )</th>
<th>Maximum weed population</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.01</td>
<td>0.0000001</td>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 6

The simulation results of Rastrigin function minimization.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Max</th>
<th>Min</th>
<th>Median</th>
<th>SD</th>
<th>( f(\bar{x}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic map</td>
<td>-1.662e−5</td>
<td>2.4195e−5</td>
<td>-3.0059e−5</td>
<td>-1.4337e−5</td>
<td>2.2130e−4</td>
<td>4.8126e−6</td>
</tr>
<tr>
<td>Sinusoidal map</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tent map</td>
<td>0.0012</td>
<td>0.0054</td>
<td>0</td>
<td>0.0033</td>
<td>8.9421e−3</td>
<td>0.00202</td>
</tr>
<tr>
<td>PSO</td>
<td>0.0039</td>
<td>0.0081</td>
<td>0.0019</td>
<td>0.0056</td>
<td>0.0542</td>
<td>0.0112</td>
</tr>
<tr>
<td>CPSO</td>
<td>1.3327e−4</td>
<td>-3.1284e−4</td>
<td>-2.0256e−5</td>
<td>1.5375e−4</td>
<td>3.0027e−4</td>
<td>9.3492e−4</td>
</tr>
<tr>
<td>IWO</td>
<td>0.0373</td>
<td>0.0494</td>
<td>0.0012</td>
<td>0.0218</td>
<td>0.0987</td>
<td>0.0555</td>
</tr>
<tr>
<td>GA</td>
<td>0.0808</td>
<td>0.0138</td>
<td>0.0056</td>
<td>0.0349</td>
<td>0.011</td>
<td>0.0739</td>
</tr>
</tbody>
</table>

(1) Initially, set the maximum and minimum value for each variable exploited in the optimization of fitness function. Chaotically distribute the pioneering seeds over the field using the chaotic maps described in Section 3.1. It is worth noting that the variables should be normalized to the range of \((0, 1)\) before applying a chaotic map. The normalization procedure is described next:

I. Transform variable \( x \) to \( \hat{x} \) confined in the data range \((0, 1)\):

\[
\hat{x} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (6)
\]

II. Apply the chaotic sequence to the transformed variable producing a new value.

III. Translate \( \hat{x} \) into the range \((x_{\text{min}}, x_{\text{max}})\):

\[
x = x_{\text{min}} + \hat{x}(x_{\text{max}} - x_{\text{min}}) \quad (7)
\]

(2) Evaluate each weed, and rank them according to their fitness in the population.

(3) Produce new seeds with respect to each weed’s ranking in the population using Eq. (1). The newly created seeds are dispersed randomly on the field with the standard deviation computed by Eq. (2).

(4) The newly generated seeds are chaotically distributed in the neighborhood of the flowering weed using one of the chaotic maps outlined in Section 3.2. If the current chaotically distributed seed has a better estimation than the previous seed, keep the new one. Otherwise, the chaotic sequence is continued. By taking advantages of the local search superiority of chaotic search, the algorithm is guaranteed to converge much faster.

(5) The seeds are ranked again, and those with lower fitness are eliminated to reach the maximum number of weeds allowed which is preset by the user.

(6) The algorithm continues at step 3 until maximum number of iterations is reached or a predetermined preciseness criterion is satisfied.

The pseudo code of the overall CIWO algorithm is given in Fig. 4.

4. Numerical results

4.1. Quadratic function

The first fitness function suggested is the simple quadratic function which is quite prevalent in nonlinear optimization problems encountered in engineering. The quadratic function is described as

\[
f(\bar{x}) = \sum_{i=1}^{10} (1 - x_i)^2 \quad (8)
\]

where the variables are restricted to \((-10, 10)\). It is obvious that the minima lies in \(f(\bar{x}) = 0\) as \(x = (1, 1, \ldots, 1)\). Fig. 5 illustrates a 3-D quadratic function. For the purpose of optimization, the parameters of the CIWO algorithm are set as illustrated in Table 1. Numerical results based on the proposed CIWO method are provided in Table 2.

4.2. Rosenbrok function

The 3-D Rosenbrok function (also known as the Rosenbrok valley or banana) is depicted in Fig. 6. This non-convex function has a special interaction among its variables which is widely used for evaluating optimization algorithms. The Rosenbrok function is given as follows

\[
f(\bar{x}) = \sum_{i=1}^{9} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right) \quad (9)
\]

where the variables can vary in the range \((-4, 4)\). Parameter selection of the CIWO algorithm is addressed in Table 3. Optimization results using CIWO are presented in Table 4.

4.3. Rastrigin function

Rastrigin’s function is a typical example of multi-modal functions which is used to test the capabilities of optimization methods. Fig. 7 shows the 3-D Rastrigin function. Equation below expresses the Rastrigin function as used in this study
The problem variables are confined to \( x \in (-5, 5) \). CIWO parameters are given in Table 5, and the simulation results are addressed in Table 6.

### 4.4. Griewangk function

Griewangk function is a continuous non-convex multimodal quadratic test function which is portrayed in Fig. 8. The function is given by the equation given below

\[
  f(x) = \sum_{i=1}^{10} \left( \frac{x_i^2}{4000} \right) - \prod_{i=1}^{10} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1
\]

in which the bound is \( x \in (-600, 600) \). As it is obvious, the minimum of the function occurs at \( x_0 = (0, 0, \ldots, 0) \). The CIWO algorithm is tuned according to the parameters given in Table 7, and optimization results are shown in Table 8.

### 4.5. Optimization problems with nonlinear constraints

In order to further illustrate the usefulness of the proposed CIWO method, the performance of the CIWO algorithm is tested against two optimization problems with nonlinear constraints from Ref. [26]. Note that these nonlinear constrained problems can be readily dealt with by adding a step which checks whether the constraint is feasible for a possible solution. The parameter selection of the CIWO algorithm for both problems provided in this section is similar to those given in Table 1.

The first problem is to minimize

\[
  f(x) = -\left( \sqrt{n} \prod_{i=1}^{n} x_i \right)
\]

Subject to

\[
  \sum_{i=1}^{n} x_i^2 - 1 = 0
\]

where \( n = 10 \) and \( 0 < x_i < 1 \) (\( i = 1, 2, \ldots, 10 \)). The global minimum is located at \( x_0 = (0.316243, 0.316243, 0.316243) \) where \( f(x) = -1.0005 \). The results of applying CIWO along with IWO, PSO, CPSO, and GA are given in Table 9.

The second problem derived from [26] is to minimize

\[
  f(x) = -(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100
\]

Subject to

\[
  (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0
\]

wherein \( 0 < x_i < 10 \) (\( i = 1, 2, 3 \)) and \( p, q, r = 3, 4, 5 \). The feasible region of search space consists of \( 4^3 \) disjoint spheres. The optimum solution is \( x_0 = (5, 5, 5) \) where \( f(x) = -1 \).

Table 10 shows the results obtained utilizing the CIWO algorithm.
Having assessed the simulation results given in this section, we adopt the sinusoidal map as the chaotic sequence generator in CIWO methodology.

5. Parameter identification of chaotic systems using CIWO algorithm

5.1. The identification algorithm

The parameter identification of chaotic systems is an active research subject [27–32]. Thus far, several invaluable investigations have been conducted by researchers suggesting the utilization of evolutionary algorithms, such as PSO, GA, COA, and etc., for identification of chaotic systems [32–38]. Consider a chaotic dynamical system defined by

\[ \dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{W}) \quad (16) \]

where \( \mathbf{x} \) denotes the state vector, \( \mathbf{W} \) represents the unknown parameter vector, and \((\cdot)^{\circ}\) is the derivative operator. The identification method based on CIWO algorithm is discussed next.

5.1.1. Preliminaries

Each weed includes a string of different values of unknown parameters. The suggested fitness function is given below:

\[ J_c = \sum_{i=0}^{N} \left[ (x_1(t) - \hat{x}_1(t))^2 + (x_2(t) - \hat{x}_2(t))^2 + \cdots + (x_M(t) - \hat{x}_M(t))^2 \right] \quad (17) \]

In which \( \hat{x}_i(t) \) is the estimated \( i \)-th state at time step \( t \). \( N \) is the number of samples from the system to be implemented for identification. \( M \) denotes the number of state variables. Note that in case of discrete-time chaotic systems the fitness function converts to

\[ J_d = \sum_{i=1}^{N} \left[ (x_1(i) - \hat{x}_1(i))^2 + (x_2(i) - \hat{x}_2(i))^2 + \cdots + (x_M(i) - \hat{x}_M(i))^2 \right] \quad (18) \]

Once the information about the states of a chaotic system is available, the identification algorithm can be readily applied. In this study, the system parameters are set in advance, and a pre-defined level of noise is added to the
Then, the system is simulated and state values are calculated for each time step.

5.1.2. The algorithm

I. To begin, a set of initial values are assigned to each parameter. Afterwards, these values are encoded into weeds. This can be done as it is described in the first step of the CIWO algorithm.

II. Following the CIWO steps, the ranking, the chaotic search and the execution stages are performed accordingly.

III. When a desired level of precision; i.e., $|J| \leq \varepsilon$ where $\varepsilon$ is very small number close to zero, or the maximum number of iteration cycles is reached the algorithm abolishes.

IV. The optimum values of parameters are extracted from the best weed which has most aptly minimized the fitness function.

5.2. Examples

5.2.1. Rössler’s system (Rössler attractor)

In 1976, Otto E. Rössler introduced a simple continuous-time dynamical system which displayed chaotic performance. The system is reportedly very helpful in modeling the equilibrium of chemical processes [39,40]. Rössler’s system of differential equations is

$$\begin{align*}
\dot{x}_1 &= -x_2 - x_3 \\
\dot{x}_2 &= x_1 + ax_2 \\
\dot{x}_3 &= b + x_1x_3 - cx_3
\end{align*}$$

The parameters $a$, $b$, and $c$ determine the system’s evolution. Originally, Rössler studied the case in which $a = 0.2$, $b = 0.2$, and $c = 5.7$; however, in this investigation $a = 0.1$, $b = 0.1$, and $c = 14$ are chosen, since this set of values are more frequently used. The noise level is chosen to be 0.01 which is defined by the noise standard deviation ratio divided by standard deviation of the noise-free system. The phase diagrams of the simulated Rössler system are given in Fig. 9. The identification algorithm as discussed in section 5.1.2. is applied. The algorithm settings and estimated parameters are provided in Tables 11 and 12, respectively.

5.2.2. Lorenz’s system (Lorenz attractor)

The Lorenz’s system was proposed by Edward Lorenz in 1963 as he was undertaking research on weather prediction. The differential equations, actually, designates a mathematical model for thermal convection. The model
involves descriptions of heat distribution, the motion of viscous fluids (atmosphere), and the driving force of thermal convection [9]. The Lorenz’s system is expressed as follows

\[
\begin{align*}
   \dot{x}_1 &= a(x_2 - x_1) \\
   \dot{x}_2 &= x_1(b - x_3) - x_2 \\
   \dot{x}_3 &= x_1x_2 - cx_3
\end{align*}
\]

The values for system parameters are listed below which ensure a chaotic behavior

\[a = 10, \quad b = 28, \quad c = 8/3\]

By setting the system parameters as given above and adding the same noise level as in the previous example, the system performance is simulated. The system trajectories are given in Fig. 10.

The CIWO algorithm as discussed is applied. Tables 13 and 14 provide the identification results and the CIWO algorithm’s parameter selection.

5.2.3. Henon’s System

The Henon System is one of the most well studied chaotic discrete-time dynamic systems [41]. The dynamics of the system are governed by

\[
\begin{align*}
   x_1(k+1) &= x_2(k) + 1 - ax_1^2(k) \quad \text{(21.a)} \\
   x_2(k+1) &= bx_1(k) \quad \text{(21.b)}
\end{align*}
\]

Henon map possess chaotic behavior as \(a = 1.4\) and \(b = 0.3\). The system is simulated and the phase space diagram of Henon map is depicted in Fig. 11. The identification results and experiment attributes are listed in Tables 15 and 16.

6. Conclusions

In this paper, a new straightforward hybrid optimization algorithm is proposed combining the COA and the IWO. The capabilities of the optimization algorithm are verified through solving a series of optimization problems using multi-dimensional benchmark functions. A significant application put forward in this study is that the proposed CIWO algorithm can be implemented for the purpose of parameter identification of chaotic systems. The results based on the proposed scheme are compared with those of the IWO, COA, PSO, CPSO, and GA, where in all cases the CIWO strategy contributes to superior estimation performance. A suggested topic for further research is
to exploit the CIWO algorithm to cope with the challenging optimization problems arise in engineering applications.

### References


