A Modified Artificial Bee Colony Algorithm for Solving Optimization Problems

Wei-Der Chang*, Shan-Cheng Pan, Ray-Jer Lee, Cheng-Hua Ku

Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan

*Corresponding Author: wdchang@stu.edu.tw

Abstract

In this paper, we develop an improved version of artificial bee colony (ABC) algorithm for solving the numerical optimization problem. In the proposed algorithm, the uniformly random number considered in the formula of generating a new food source is replaced by the chaotic random number which is provided from the Chen’s system. To show the feasibility of the proposed scheme, a function minimization problem with constrained conditions is simulated using a large number of different sets of initial conditions. It is concluded from simulation results and comparisons that the proposed method is superior to the general ABC algorithm.

Keywords: chaotic random numbers, artificial bee colony (ABC) algorithm, Chen’s system.

1. Introduction

The commonly used random distributions are the uniform distribution and the normal distribution. The former means that the occurrence probability for each point is all the same, and the latter is that the highest occurrence probability occurs in the mean point and then it decays symmetrically around this point. For most of engineering applications, these two kinds of random distributions are very important and often taken into account. However, this paper will discuss the chaotic random distribution and apply it into the artificial bee colony (ABC) algorithm. It is well known that chaos behavior frequently occurs in some highly nonlinear systems and it has certain features including hyper sensitivity to initial values, unpredictable system states, board spectra for its Fourier transform, and fractal properties of the motion in the phase plane. About the chaos research, there are many published articles such as bifurcation analysis [1], chaos control and synchronization [2], fractional order system [3], and geometrical analysis onset of chaos [4] etc. In this study, the Chen’s system is studied and will take the system state to be the chaotic random number. A uniformly random number used in the update formula of ABC algorithm is then replaced by the chaotic random number.

Artificial bee colony (ABC) algorithm was initially developed by Karaboga and his research team [5-7]. It is stimulated from the social behavior of honey bee colony. In the ABC algorithm, the colony of artificial bees is divided into three kinds of bees including employed bees, onlookers, and scouts [5]. Each of bee colonies plays its specific role and has the corresponding action to achieve the optimization design for certain optimized problem. In addition, The ABC is also a population-based algorithm as well as other evolutionary computations, in which the candidate solution is called the food source. Based on using the ABC algorithm, a variety of engineering applications were solved and discussed in recent years [8-10]. For instance, a hybrid algorithm combining ABC and bees algorithm (BA) was proposed by Tsai [8], and several constrained optimization problems were successfully solved. In [9], they presented an application of ABC algorithm to solve a series-parallel system availability allocation problem for butter-oil processing plant. In [10], the authors presented the implementation of multi-objective based optimization of ABC algorithm for load frequency control on a two area interconnected reheat thermal power system. Numerical results revealed that the proposed scheme is robust enough to operate under different operating conditions and system parameter variations.

In this paper, an improved version of ABC algorithm with chaotic random numbers is developed, and based on
the proposed algorithm a function minimization problem with constrained conditions is solved. The rest of this paper is organized as follows. In Section II, the clear description for Chen chaotic system is addressed and this system is considered as a generator of chaotic random numbers. A modified ABC with chaotic random numbers is introduced in Section III. Section IV will show simulation results and comparisons with the general ABC to confirm the applicability. Finally, a brief conclusion is stated in Section V.

2. Chaotic Random Numbers

In this study, the chaotic random number is generated by Chen’s system which is described by the following state equations [11]

\[
\dot{x} = 35(y-x), \quad (1a)
\]

\[
\dot{y} = -7x - xz + 3y, \quad (1b)
\]

\[
\dot{z} = xy - 28z, \quad (1c)
\]

where \(x, y, \) and \(z\) denote the system state variables. Fig. 1 clearly shows the chaos dynamics of Chen’s system. Such a chaotic system is considered to generate a series of chaotic random numbers. To achieve that, Eq. (1) is requested to be converted into a discrete form by using the Euler’ method as

\[
x(k) = x(k-1) + T \cdot 35(y(k-1) - x(k-1)), \quad (2a)
\]

\[
y(k) = y(k-1) + T \cdot (-7x(k) - x(k)z(k) + 3y(k)), \quad (2b)
\]

\[
z(k) = z(k-1) + T \cdot (x(k)y(k) - 28z(k)), \quad (2c)
\]

where \(T\) is the sampling interval, and here it is set to \(T = 0.0005\) for simulations. Moreover, the initial values of three system states are given by \(x(0) = 10.0\), \(y(0) = 10.0\), and \(z(0) = 10.0\), respectively, and the total of the sampling number is set to be 20000, i.e., the simulating time is 10 seconds. Fig. 2 then shows their state trajectories of \(x, y, \) and \(z\). As a result, we have obtained 20000 data including state variables, \(x, y, \) and \(z\), respectively. Any system state is able to be taken as the chaotic random number generator and here the chaotic state \(x\) is considered for simulations in this study. On the other hand, for most physical applications the commonly used random number is constrained by certain interval, for instance, the interval \([-1, 1]\). Accordingly, a series of \(x\) random chaotic numbers obtained from Eq. (2) are requested to be condensed by the following formula

\[
x_{\text{new}} = 2 \cdot \frac{x_{\text{old}} - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} - 1,
\]

where \(x_{\text{old}}\) is the original state value, \(x_{\text{new}}\) is a condensed value, \(x_{\text{max}}\) and \(x_{\text{min}}\) denote the maximum and minimum values of chaotic state \(x\), respectively. After doing this, all the condensed values by Eq. (3) are within the interval \([-1, 1]\).

3. A Modified ABC Algorithm

Karaboga and his research team first developed the ABC algorithm and have applied it into a variety of engineering optimization problems. The concept of ABC algorithm is basically inspired by the organizational behavior of honeybee colony. It has three different kinds of bee colonies: the employed bees, onlookers, and scouts.
The employed bee is defined as a bee that is going to the food source previously visited by itself, and a bee which is waiting on the dance area for making decision to choose a food source is called the onlooker, and finally the scout bee means that a bee is carrying out random search for food sources. First half of the colony is composed of employed artificial bees and the remainder half then contains onlooker bees. Every employed bee corresponds to one food source and this means that the number of employed bees is equal to the number of food source. Moreover, an employed bee will become a scout when its food source is exhausted by itself and onlooker bees [5][12].

Each food source represents a candidate solution of the optimized problem in the ABC algorithm. Here let $X_i = [x_{i1}, x_{i2}, \cdots, x_{iN}]$ be the $i$th food source containing $N$ components, where $x_{i1}, x_{i2}, \cdots, x_{iN}$ are the design variables. To evaluate the performance of each food source, the cost function (amount of nectar) denoted by $CF(X_i)$ is calculated. According to this cost function, a probability value of $P(X_i)$ for the $i$th food source $X_i$ is then evaluated in order to determine whether an onlooker bee chooses a new food source. For every $X_i$, the probability is defined by

$$P(X_i) = \frac{CF(X_i)}{\sum_{n=1}^{H} CF(X_n)} \quad \text{for} \quad i = 1, 2, \cdots, H$$

(4)

where $CF(X_i)$ is the cost function of the $i$th food source $X_i$ and $H$ is the number of the food sources, i.e., population size. It is clear from Eq. (4) that the probability is proportional to its corresponding cost function. Moreover, the formula to produce a new food source candidate around itself is very important in the ABC algorithm, and it is given by Eq. (5)

$$x_{ij}^{new} \leftarrow x_{ij} + \phi_j (x_{ij} - x_{kj}) \quad \text{for} \quad i = 1, 2, \cdots, H \quad \text{and} \quad j = 1, 2, \cdots, N$$

(5)

where $x_{ij}^{new}$ is the new candidate position, $\phi_j$ is a uniformly random number chosen from some interval for the general ABC, and $k$ is a random integer selected from $\{1, 2, \cdots, H\}$ but $k \neq i$. The ABC algorithm uses Eq. (5) to achieve the system optimization. In this study, $\phi_j$ is modified to the chaotic random number for optimization search, which has been addressed in Section II. Some comparisons between the proposed and the original algorithms will be given in next section.

Design steps for the modified ABC algorithm with chaotic random numbers can be outlined in the following for solving optimization problem.

Step 1. Generate an initial population with $H$ food sources randomly.

Step 2. If the number of generations $G$ is achieved, then the algorithm stops.

Step 3. Evaluate the cost function $CF(X_i)$ and the probability $P(X_i)$ for every food source $X_i$.

Step 4. Create a random number $p_i$ from the interval $[0, 1]$. If $P(X_i) \leq p_i$, the formula to produce a new candidate $X_i^{new}$ is performed using Eq. (5) with chaotic random numbers for $i$th food source $X_i$.

Step 5. Evaluate the cost function $CF(X_i^{new})$ and compare with $CF(X_i)$. If $CF(X_i^{new}) < CF(X_i)$, the new candidate $X_i^{new}$ then replaces the original $X_i$; otherwise, the food source $X_i$ remains.

Step 6. When the number of generations is equal to the value, named limit, then check whether the positions of every $X_i$ have been changed. If none, this food source is discarded and the algorithm reproduces a new food source form searching space uniformly.

Step 7. Go back to Step 2.

4. Optimization Problem with Constrained Conditions

In this section, a function minimum problem with some constrained conditions is simulated by using the
proposed method. Parameter settings of the algorithm are given by the population size $H = 20$, number of generations $G = 2000$, and $limit = 40$, respectively. The perturbing random number $\phi_j$ in Eq. (5) is uniformly chosen from the interval $[-0.1, 0.1]$ for the general ABC. Hence, in the proposed method the chaotic random number generated by Eq. (3) is requested to be further multiplied by 0.1 in order to meet this interval for fair comparisons. A function that is considered to be minimized is given by [13]

$$f(X) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2,$$

subject to

$$0 \leq x_1 \leq 6, \quad 0 \leq x_2 \leq 6,$$ and

$$g_1(X) = 4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \geq 0,$$

$$g_2(X) = x_1^2 + (x_2 - 2.5)^2 - 4.84 \geq 0,$$

where $X = [x_1, x_2]$ represents the food source which will be designed. This is a constrained minimization problem consisting of two decision variables $x_1$ and $x_2$, and two inequality conditions. The optimal system solution occurs at $x^*_1 = 2.246826$ and $x^*_2 = 2.381865$ implying $f(X^*) = 13.590839$. For comparisons, ten runs with different sets of initial conditions are examined for the proposed and general ABC algorithms. Numerical simulation results are listed in Tabs. 1 and 2, respectively. Tab.1 is for the general ABC algorithm and Tab. 2 for the proposed algorithm. It is concluded from Tabs. 1 and 2 that the proposed algorithm is slightly better than the general ABC algorithm and the robustness of the algorithm is also guaranteed.

5. Conclusions

In this paper, a modified ABC algorithm has successfully been developed for solving the optimization problem. Instead of the uniformly random number, the chaotic random number obtained from Chen’s system is utilized in the update formula for the new food source. Many runs with different initial conditions are executed to verify the applicability of the proposed method. Simulation results reveal that the modified ABC algorithm outperforms the general ABC.

<table>
<thead>
<tr>
<th>Run</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(X)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.246812</td>
<td>2.381615</td>
<td>13.590847</td>
</tr>
<tr>
<td>2</td>
<td>2.246811</td>
<td>2.381589</td>
<td>13.590846</td>
</tr>
<tr>
<td>3</td>
<td>2.246812</td>
<td>2.381615</td>
<td>13.590847</td>
</tr>
<tr>
<td>4</td>
<td>2.246811</td>
<td>2.381589</td>
<td>13.590846</td>
</tr>
<tr>
<td>5</td>
<td>2.246812</td>
<td>2.381615</td>
<td>13.590847</td>
</tr>
<tr>
<td>6</td>
<td>2.246811</td>
<td>2.381589</td>
<td>13.590846</td>
</tr>
<tr>
<td>7</td>
<td>2.246812</td>
<td>2.381615</td>
<td>13.590847</td>
</tr>
<tr>
<td>8</td>
<td>2.246811</td>
<td>2.381589</td>
<td>13.590846</td>
</tr>
<tr>
<td>9</td>
<td>2.246812</td>
<td>2.381615</td>
<td>13.590847</td>
</tr>
<tr>
<td>10</td>
<td>2.246811</td>
<td>2.381589</td>
<td>13.590846</td>
</tr>
</tbody>
</table>

**mean** 2.246815 2.381602 13.5908465

**optimal solution** 2.246826 2.381865 13.590839

<table>
<thead>
<tr>
<th>Run</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(X)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.246829</td>
<td>2.381916</td>
<td>13.590847</td>
</tr>
<tr>
<td>2</td>
<td>2.246813</td>
<td>2.381621</td>
<td>13.590845</td>
</tr>
<tr>
<td>3</td>
<td>2.246829</td>
<td>2.381916</td>
<td>13.590847</td>
</tr>
<tr>
<td>4</td>
<td>2.246813</td>
<td>2.381621</td>
<td>13.590845</td>
</tr>
<tr>
<td>5</td>
<td>2.246829</td>
<td>2.381916</td>
<td>13.590847</td>
</tr>
<tr>
<td>6</td>
<td>2.246813</td>
<td>2.381621</td>
<td>13.590845</td>
</tr>
<tr>
<td>7</td>
<td>2.246829</td>
<td>2.381916</td>
<td>13.590847</td>
</tr>
<tr>
<td>8</td>
<td>2.246813</td>
<td>2.381621</td>
<td>13.590845</td>
</tr>
<tr>
<td>9</td>
<td>2.246829</td>
<td>2.381916</td>
<td>13.590847</td>
</tr>
<tr>
<td>10</td>
<td>2.246813</td>
<td>2.381621</td>
<td>13.590845</td>
</tr>
</tbody>
</table>

**mean** 2.246821 2.3817685 13.590846

**optimal solution** 2.246826 2.381865 13.590839

**Acknowledgment**

This work was supported by the Ministry of Science and Technology of Taiwan under the grant MOST 103-2221-E-366-004.

**References**

(1) E.M. Elabbasy, A.A. Elsadany, Y. Zhang, Bifurcation


(8) H.C. Tsai, Integrating the artificial bee colony and bees algorithm to face constrained optimization problems, Information Sciences 258 (2014) 80-93.


