Entropy **2015**, *17*, 692-709; doi:10.3390/e17020692

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

Parameters Estimation of Uncertain Fractional-Order Chaotic Systems via a Modified Artificial Bee Colony Algorithm

Wei Hu [1], Yongguang Yu [1,*] and Sha Wang [2]

[1] School of Science, Beijing Jiaotong University, Beijing 100044, China;
E-Mail: 13121527@bjtu.edu.cn

[2] School of Information, Beijing Wuzi University, Beijing 101149, China;
E-Mail: saya1016@yahoo.cn

* Author to whom correspondence should be addressed; E-Mail: ygyu@bjtu.edu.cn;
Tel.: +86-10-51684135.

Academic Editors: Guanrong Chen, C.K. Michael Tse, Mustak E. Yalcin, Hai Yu and Mattia Frasca

Received: 25 December 2014 / Accepted: 27 January 2015 / Published: 4 February 2015

Abstract: Parameter estimation for fractional-order chaotic systems has been an interesting and important issue in theory and various fields of application. In this paper, fractional orders, as well as systematic parameters of fractional-order chaotic systems are considered by treating fractional orders as additional parameters. The parameter estimation is transformed into a multidimensional optimization problem, and an effective modified artificial bee colony algorithm is proposed to solve this problem. Numerical simulations are conducted on two typical fractional-order chaotic systems to show the effectiveness of the proposed scheme.

Keywords: parameter estimation; uncertain fractional-order chaotic systems; modified artificial bee colony algorithm

---

1. Introduction

Recently, considerable attention has been give to making use of the great potential of fractional calculus in physics [1], electrical circuit theory [2] and control systems [3]. In particular, a significant role is played in chaos control theory, and many control methods have been devised for fractional-order chaotic systems under the condition of known fractional orders and systematic parameters. However, in the real world, the fractional orders and systematic parameters of fractional-order chaotic systems cannot be exactly known. Therefore, to achieve the goal of the control and synchronization of fractional-order chaotic systems, the parameters should be estimated beforehand if they are unknown.

However, direct measurement of parameters for the fractional-order chaotic systems is very difficult, so estimating system parameters through an observed chaotic scalar time series based on parameter estimation methods is feasible and valuable. Recently, by formulating the problem as a multidimensional optimization problem and treating the fractional order as one of the parameters to be estimated, many optimization approaches have been applied to solve this problem, such as the particle swarm optimization (PSO) [4–6] and differential evolution (DE) [7–9].

The artificial bee colony (ABC) algorithm is one of the most recently introduced swarm-based optimization algorithms. It was developed by Karaboga in 2005 based on simulating the foraging behavior of a honeybee colony and has been shown to be competitive with other population-based algorithms for the global numerical optimization problem with the advantage of employing fewer control parameters [10–15]. For instance, apart from the maximum evaluation number and population size, a standard GA has three more control parameters (crossover rate, mutation rate, generation gap) [16], a standard DE has at least two control parameters (crossover rate, scaling factor) [17] and a basic PSO has three control parameters (cognitive and social factors, inertia weight) [4]. Besides, limit values for the velocities v_{max} have a significant effect on the performance of PSO. The ABC algorithm has only one control parameter ($limit$) apart from colony size and maximum cycle number. Although it uses less control parameters, the performance of the ABC algorithm is better than or similar to that of these algorithms, and it can be efficiently used for solving multimodal and multidimensional optimization problems.

Similar to other population-based algorithms, the ABC algorithm can be further improved to enhance the exploration and exploitation abilities. Therefore, in this paper, a modified artificial bee colony (MABC) algorithm is put forward. Additionally, it is modified by employing two new searching equations. The MABC algorithm is further used to estimate the parameters of uncertain fractional-order chaotic systems via a functional extrema model. Numerical simulations are performed to estimate two well-known fractional-order chaotic systems and statistically compared with some typical existing methods. The simulation results demonstrate the good performance and the superiority of the MABC algorithm, and thus, the MABC algorithm proves to be a promising candidate for parameter estimation of uncertain fractional-order chaotic systems.

The rest of the paper is organized as follows. In Section 2, some preliminaries are given. In Section 3, a brief problem formulation is described. Section 4 describes the MABC algorithm in sufficient detail. The simulation results based on the MABC algorithm are given in Section 5, and the compared results based on PSO and DE are also presented. The paper ends with conclusions in Section 6.

2. Preliminaries

2.1. Caputo Fractional-Order Derivative

There are several definitions of fractional-order derivatives. There are three best-known definitions: the Grunwald–Letnikov, Riemann–Liouville and Caputo definitions [18]. In particular, the Caputo fractional-order derivative has the same initial conditions as integer-order derivatives, which is well understood in physical situations and more applicable to real-world problems. Thus, the Caputo fractional-order derivative is introduced in this paper.

Definition 1. *Caputo fractional-order derivative: The Caputo fractional-order derivative of order* $\alpha > 0$ *for a function* $f(t) \in C^{n+1}([t_0, +\infty), R)$ *is defined as:*

$$ {}_{t_0}D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_{t_0}^{t} \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau, \tag{1}$$

where $\Gamma(\cdot)$ *denotes the gamma function and* n *is a positive integer, such that* $n - 1 < \alpha \leq n$.

2.2. The Standard Artificial Bee Colony (ABC) Algorithm

The standard artificial bee colony (ABC) algorithm proposed by Karaboga in 2005 is a competitive optimization technique that simulates the intelligent foraging behavior of a honeybee colony [10–12]. In fact, the ABC algorithm is an iteration optimization technique similar to other population-based algorithms. In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlooker bees and scouts. The first half of the colony consists of the employed artificial bees, and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of food sources is equal to the number of employed bees. The employed bee of an abandoned food source becomes a scout. The search carried out by the artificial bees can be described as follows.

2.2.1. Initialization of the Population

The initial population of solutions is filled with SN number of n-dimensional real-valued vectors, which are generated randomly (*i.e.*, food sources). Let $X_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$ represent the i-th food source in the population, and then, each food source is generated as follows:

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j}), \tag{2}$$

where $i = 1, 2, \cdots, SN$, $j = 1, 2, \cdots, D$. $x_{min,j}$ and $x_{max,j}$ are the lower and upper bounds for the dimension j, respectively. These food sources are randomly assigned to SN number of employed bees, and their fitness is evaluated accordingly.

2.2.2. The Employed Bee Phase

At this stage, for each position of the employed bee's food source X_i, a new food source position V_i is generated via the equation as follows:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}), \tag{3}$$

where $k = 1, 2, \cdots, SN$ and $j = 1, 2, \cdots, D$. k and j are randomly generated, and k must be different from i. $\phi_{i,j}$ is a random number in $[-1, 1]$. The above explanation implies that the other components of V_i, except for dimension j, are the same as the ones of X_i.

Then, a greedy selection is made between X_i and V_i. That is, once V_i is obtained, it will be evaluated and compared with X_i. If the fitness of V_i is equal to or better than that of X_i, consequently, V_i will replace X_i and correspondingly become a member of the population. Otherwise, X_i is retained.

2.2.3. Calculating Probability Values Referring to the Probability Selection

After finishing the update process, employed bees share their nectar amount information, which is related to the food sources, with the onlooker bees in the nearby hive. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source through a probability value p_i, which is calculated by the following form:

$$p_i = \frac{fit_i}{\sum\limits_{j=1}^{SN} fit_j}, \tag{4}$$

where fit_i denotes the fitness value of solution X_i. It is obvious that the higher the fitness value of solution X_i, the higher the probability of selecting the i-th food source corresponding to solution X_i is. Besides, the fitness value fit_i is defined as follows:

$$fit_i = \begin{cases} \frac{1}{1+f(X_i)}, & if\ f(X_i) \geq 0, \\ 1 + |f(X_i)|, & if\ f(X_i) < 0, \end{cases} \tag{5}$$

where $f(X_i)$ represents the objective function value of the decision vector X_i.

2.2.4. The Onlooker Bee Phase

Based on the probability value p_i calculated by Equation (4), each onlooker bee randomly chooses a food source corresponding to the solution X_i with a probability value p_i. Thereafter, it makes a modification (*i.e.*, V_i) around the chosen food source by the Equation (3). Similarly to the phase of employed bees, the greedy selection mechanism is applied to select a better solution between solution X_i and V_i.

2.2.5. The Scout Bee Phase

In this phase, the number trial denotes the times that the food source position corresponding to the solution X_i is not improved continuously in the honeybee's memory. Additionally, once the value of the number trial is more than the predetermined parameter limit employed in the ABC algorithm, the corresponding employed bee abandons the food source and becomes a scout bee. The scout bee produces a new food source randomly as Equation (2).

2.2.6. Framework of the Standard Artificial Bee Colony Algorithm

Based on the above introduction, the framework of the standard artificial bee colony algorithm can be given as below:

Algorithm 1: Framework of the standard artificial bee colony algorithm.

Step (0) Predefine some parameters: SN (population size), D (the dimension of optimization problem), $LOWER$ (the lower bound of searching space), $UPPER$ (the upper bound of searching space), $limit$ (control parameter), MCN (maximum cycle number).

Step (1) The population initialization phase:

Step (1.1) Randomly generate $0.5 * SN$ points in the searching space to form an initial population via Equation (2).

Step (1.2) Evaluate the objective function values of population.

Step (1.3) $cycle = 1$;

Step (2) The employed bee's phase:

For $i = 1$ **to** $0.5 * SN$ **do**

Step (2.1)

Step (2.1.1) Generate a candidate solution V_i by Equation (3).

Step (2.1.2) Evaluate $f(V_i)$.

Step (2.2) If $f(V_i) < f(X_i)$, set $X_i = V_i$, otherwise, set $trial_i = trial_i + 1$.

End For

Step (3) Calculating the probability values p_i by Equation (4), set $t = 0, i = 1$.

Step (4) The onlooker bee's phase:

While $t \leq 0.5 * SN$, **do**

Step (4.1)

If $rand(0, 1) < p_i$

Step (4.1.1) Generate a candidate solution V_i by Equation (3).

Step (4.1.2) Evaluate $f(V_i)$.

Step (4.1.3) If $f(V_i) < f(X_i)$, set $X_i = V_i$, otherwise, set $trial_i = trial_i + 1$.

Step (4.1.4) Set $t = t + 1$.

End If

Step (4.2) Set $i = i + 1$, if $i = 0.5 * SN$, set $i = 1$.

End While

Step (5) The scout bee's phase:

If $max(trial_i) > limit$, replace X_i with a new candidate solution generated via Equation (2).

Step (6) Set $cycle = cycle + 1$, and **if** $cycle > MCN$, then stop and output the best solution achieved so far, otherwise, go to **Step 2**.

3. Problem Formulation

Consider the following fractional-order chaotic system:

$$ {}_0D_t^{\alpha} Y(t) = f(Y(t), Y_0, \theta), \tag{6} $$

where $Y(t) = (y_1(t), y_2(t), \cdots, y_n(t))^T \in R^n$ denotes the state vector of System (6), $Y_0 = Y(0)$ denotes the initial value, $\theta = (\theta_1, \theta_2, \cdots, \theta_n)^T$ denotes the set of original systematic parameters and $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)(0 < \alpha_i < 1, i = 1, 2, \cdots, n)$ are the fractional derivative orders. Additionally, $f(Y(t), Y_0, \theta) = (f_1(Y(t), Y_0, \theta), f_2(Y(t), Y_0, \theta), \cdots, f_n(Y(t), Y_0, \theta))^T$.

The corresponding estimated system can be written as:

$$ {}_0D_t^{\tilde{\alpha}} \tilde{Y}(t) = f(\tilde{Y}(t), Y_0, \tilde{\theta}), \tag{7} $$

where $\tilde{Y}(t) = (\tilde{y}_1(t), \tilde{y}_2(t), \cdots, \tilde{y}_n(t))^T \in R^n$ is the state vector of the estimated System (7), $\tilde{\theta}=(\tilde{\theta}_1, \tilde{\theta}_2, \cdots, \tilde{\theta}_n)^T$ is a set of estimated systematic parameters and $\tilde{\alpha} = (\tilde{\alpha}_1, \tilde{\alpha}_2, \cdots, \tilde{\alpha}_n)^T$ is the estimated fractional orders. Besides, Systems (6) and (7) have the same initial conditions Y_0.

In general, to identify the fractional-order chaotic System (6), it can be transformed into a functional extrema model as follows:

$$ \begin{aligned} J(\tilde{\alpha}, \tilde{\theta}) &= arg \min_{(\tilde{\alpha}, \tilde{\theta}) \in \Omega} F \\ &= arg \min_{(\tilde{\alpha}, \tilde{\theta}) \in \Omega} \sum_{k=1}^{N} \| Y_k - \tilde{Y}_k \|_2, \end{aligned} \tag{8} $$

where $k = 1, 2, \cdots, N$ is the sampling time point and N denotes the length of data used for parameter estimation. Y_k and $\tilde{Y}_k$ respectively denote the state vector of the original System (6) and the estimated System (7) at time kh. h is the step size employed in the predictor-corrector approach for the numerical solutions of fractional differential equations [19]. $\| \cdot \|$ is the Euclid norm. Ω is the searching area admitted for parameters $\tilde{\theta}$, where fractional orders $\tilde{\alpha}$ are considered as special variables. The parameter identification of System (6) can be achieved by searching suitable $\tilde{\theta}$ and $\tilde{\alpha}$ in the searching space Ω, such that the objective function is minimized. In other words, the main task is to find the best combination of the independent variables of $\tilde{\alpha}$ and $\tilde{\theta}$ for the objective function.

4. A Modified Artificial Bee Colony Algorithm

In order to improve the convergence speed and accuracy of the standard ABC algorithm, some modifications in terms of the searching equations are proposed. Additionally, they are described in detail in the following subsections.

4.1. Two Modified Solution Searching Equations

To improve the performance of the standard ABC algorithm, one effective research trend is to investigate searching equations. So far, a variety of searching equations have been put forward [20–24]. The most representative one is GABCproposed by Zhu and Kwong [21]. In this method, in order to improve the exploitation and make full use of the information of the gbest solution to guide the search of candidate solution, a modified searching equation enlightened by PSO is suggested as follows:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) + \psi_{i,j}(x_{best,j} - x_{i,j}), \tag{9}$$

where the third term in the right-hand side of Equation (9) is a new added term called the gbest term, $x_{best,j}$ is the j-th element of the global best solution and $\psi_{i,j}$ is a uniform random number in $[0, 1.5]$. Nevertheless, based on the experiment results shown in [21], it can be found that the improvement of the algorithm is not notable.

Thus, owing to the poor performance of the above Equation (3) and Equation (9), two new searching equations are presented to be used by the employed and onlooker bees, separately. The two equations are written as below:

$$v_{i,j} = x_{r1,j} + \lambda(x_{i,j} - x_{r1,j}) + \mu(x_{best,j} - x_{r1,j}), \tag{10}$$

$$v_{i,j} = x_{best,j} + p_i(x_{i,j} - x_{best,j}) + \nu(x_{r1,j} - x_{r2,j}), \tag{11}$$

where $x_{best,j}$ is the j-th element of the global best solution. Similar to PSO algorithms, $x_{i,j}$ denotes the j-th element of the best previous solution of employed bee i (*i.e.*, the local-best position or its experience). $r1$ and $r2$ are distinct integers randomly selected from $\{1, 2, \cdots, SN\}$ and are also different from i and $best$. $j \in \{1, 2, \cdots, n\}$ is a randomly chosen index. p_i represents the current probability of the i-th employed bee. λ, μ are a random number in the range $[0, 1]$, and ν is a random number in the range $[-1, 1]$.

From Equation (10), due to the first term, the new candidate solution is generated around X_{r1}, which is a randomly selected individual from the population. The randomly selected individual X_{r1} can bring more information to the searching equation and avoid being trapped into the local optimum, which is good at exploration. Besides, despite the guidance of X_{best} not being used in the first term, Equation (10) can also take full advantage of $x_{i,j}$ and X_{best} in the latter two terms to drive the new candidate solution towards the current best solution, which can guarantee its convergence speed. Therefore, the last two terms of Equation (10) do well in the exploitation. Generally speaking, Equation (10) can keep the exploration and exploitation well balanced. However, because of the guidance of X_{r1} in the first term, Equation (10) pays more attention to the exploration. According to [10], in the employed bee phase, the searching process focuses on the exploration, which is consistent with the emphasis of Equation (10). Therefore, Equation (10) is chosen as the searching equation for the employed bees.

In Equation (11), owing to the first term, the new candidate solution is generated around X_{best}, which stands for the current global best solution. The current probability is introduced in the second term to make full use of the information of the current population to further enhance the exploitation. Thus, the first two terms have effects on the exploitation ability. In addition, to keep balance corresponding to the first two terms in Equation (11), the third term is recommended to improve the diversity of the population. To sum up, Equation (11) may be beneficial to not only the convergence, but also the diversity of the population. However, due to the guidance of X_{best} in the first term, Equation (11) places more emphasis on the exploitation. According to [10], the onlooker bee phase concentrates on the exploitation, which is inconsistent with the emphasis of Equation (11). Therefore, Equation (11) is selected as the searching equation for the onlooker bees.

4.2. The Proposed Method

In view of the above, a new algorithm is proposed. Two new searching equations are proposed to generate new candidate solutions on the employed bee's phase and onlooker bee's phase, respectively. The pseudo-code of the modified ABC algorithm (here, we call it MABC) is given below:

Algorithm 2: Framework of the modified ABC algorithm.

Step (0) Predefine some parameters: SN, D, $LOWER$, $UPPER$, $limit$, MCN.
Step (1) The population initialization phase:
  Step (1.1) Randomly generate $0.5 * SN$ points in the search space to form an initial population via Equation (2).
  Step (1.2) Evaluate the objective function values of the population.
  Step (1.3) $cycle = 1$;
Step (2) The employed bees phase:
  For $i = 1$ **to** $0.5 * SN$ **do**
    Step (2.1)
      Step (2.1.1) Generate a candidate solution V_i by **Equation (10)**.
      Step (2.1.2) Evaluate $f(V_i)$.
    Step (2.2) If $f(V_i) < f(X_i)$, set $X_i = V_i$, otherwise, set $trial_i = trial_i + 1$.
  End For
Step (3) Calculating the probability values p_i by Equation (4), set $t = 0, i = 1$.
Step (4) The onlooker bees phase:
  While $t \leq 0.5 * SN$, **do**
   Step (4.1)
    If $rand(0, 1) < p_i$
      Step (4.1.1) Generate a candidate solution V_i by **Equation (11)**.
      Step (4.1.2) Evaluate $f(V_i)$.
      Step (4.1.3) If $f(V_i) < f(X_i)$, set $X_i = V_i$, otherwise, set $trial_i = trial_i + 1$.
      Step (4.1.4) Set $t = t + 1$.
    End If
   Step (4.2) Set $i = i + 1$, if $i = 0.5 * SN$, set $i = 1$.
  End While
Step (5) The scout bee's phase:
  If $max(trial_i) > limit$, replace X_i with a new candidate solution generated via Equation (2).
Step (6) Set $cycle = cycle + 1$, and **if** $cycle > MCN$, then stop and output the best solution achieved so far; otherwise, go to **Step 2**.

5. Simulations

To verify the effectiveness of the MABC algorithm, two typical fractional-order chaotic systems are chosen to test the performance. The simulations were done using MATLAB (Version 7.1, MathWorks, Natick, MA, USA) on an Intel(R) Core(TM) i5-3470 CPU, 3.2 GHz with 4 GB of RAM. In order to calculate the objective function, the number of samples is set as 300 and the step size is 0.01. As is well known, the larger the population and the maximum cycle number, the larger the probability of finding the global optimum. However, a larger population and maximum cycle number implies a larger number of function evaluations. In the following simulations, for the MABC, ABC, PSO and DE algorithms, the population size (SN) and maximum cycle number (MCN) are all set as: $SN = 100$, $MCN = 100$. Besides, for the MABC and ABC algorithms, according to [10], to make a fair comparison and have good searching ability, the control parameter $limit$ is chosen as 15. For the parameters of the DE algorithm, according to [7], the scaling factor $F = 0.5$ is usually a good choice. If the parameter F is smaller, the population may converge prematurely. On the contrary, the convergence speed decreases. Besides, regarding the crossover rate CR, a large CR often speeds up the convergence. However, from a certain value upwards, the convergence speed may decrease or the population may also converge prematurely. A good choice for the crossover rate is a value between $CR = 0.3$ and $CR = 0.9$. In the following experiments, the parameters of DE are selected as: $F = 0.5$, $CR = 0.7$. For PSO, empirical results have shown that an inertia weight $w = 0.7298$ and cognitive and social factors with $c1 = c2 = 1.49618$ provide good convergent behavior [4,25]. Therefore, in the following simulation, the parameters for PSO are set as: $w = 0.7298$, $c1 = c2 = 1.49618$. To make a fair comparison, the searching spaces of the parameters are the same for all algorithms.

Example 1. *The fractional-order economic system [26,27] is described as:*

$$\begin{cases} {}_0D_t^{\alpha_1} x(t) = z(t) + (y(t) - a)x(t), \\ {}_0D_t^{\alpha_2} y(t) = 1 - by(t) - x^2(t), \\ {}_0D_t^{\alpha_3} z(t) = -x(t) - cz(t), \end{cases} \tag{12}$$

when $(a, b, c) = (1, 0.1, 1)$, $(\alpha_1, \alpha_2, \alpha_3) = (0.90, 0.85, 0.95)$ *and the initial point is* $(2, -1, 1)$, *System (12) is chaotic. To show the performance of the MABC algorithm clearly, the true values of fractional orders* $\alpha_1, \alpha_2, \alpha_3$ *and systematic parameter* a *are assumed as unknown parameters, which need to be estimated. The searching spaces of the unknown parameters and orders are set as* $(\alpha_1, \alpha_2, \alpha_3, a) \in [0.1, 1.4] \times [0.1, 1.4] \times [0.1, 1.4] \times [0.5, 1.5]$.

The statistical results of the best, the mean and the worst estimated parameters with the corresponding relative error values over 15 independent runs are shown in Table 1. From Table 1, it can be easily seen that the estimated value generated by the MABC algorithm is closer to the true parameter value, which means that it is more accurate than the standard ABC, PSO and DE algorithms. Besides, it can also be clearly found that the relative error values obtained by the MABC algorithm and marked with black are all smaller than those of the standard ABC, PSO and DE algorithms, which can further prove that the MABC algorithm has higher calculation accuracy. What is more, the best fitness value obtained by the MABC algorithm is better than those obtained by the standard ABC, PSO and DE algorithms.

The evolutionary curves of the parameters and fitness values estimated by the various algorithms are shown in Figures 1–3 in a single run. From Figures 1–3, it can be clearly seen that convergence processes of the parameters and fitness values of MABC algorithm are much better than other algorithms. The estimated parameters can be closer to the true values than the standard ABC, PSO and DE algorithms. Additionally, the relative error values and fitness values obtained by the MABC algorithm decline faster than the other algorithms. In other words, it can be concluded that the MABC algorithms can more efficiently identify a fractional-order economic system than the standard ABC algorithm, as well as the existing PSO and DE algorithms.

Table 1. Simulation results of various algorithm for System (12) over 15 independent runs. MABC, modified ABC; DE, differential evolution.

Algorithm		ABC	MABC	PSO	DE
Best	α_1	0.89999710751139	0.90000000000779	0.90000008734044	0.90000000839266
	$\frac{\lvert\alpha_1-0.90\rvert}{0.90}$	0.00000289	**0.00000000000779**	0.0000000873	0.0000000839
	α_2	0.85000340425767	0.84999999974820	0.84999849834607	0.85000001554904
	$\frac{\lvert\alpha_2-0.85\rvert}{0.85}$	0.00000340	**0.000000000252**	0.00000150	0.0000000155
	α_3	0.95000109556663	0.95000000006606	0.94999815994424	0.94999999728649
	$\frac{\lvert\alpha_3-0.95\rvert}{0.95}$	0.00000110	**0.0000000000661**	0.00000184	0.00000000271
	a	1.00000824446713	1.00000000001685	0.99999940889994	0.99999991725828
	$\frac{\lvert a-1.00\rvert}{1.00}$	0.00000824	**0.0000000000169**	0.000000591	0.0000000827
	F	0.000109	0.00000000381	0.0000496	0.00000150
Mean	α_1	0.89999898839006	0.90000000009692	0.90000391181321	0.90000002685242
	$\frac{\lvert\alpha_1-0.90\rvert}{0.90}$	0.00000101	**0.0000000000969**	0.00000391	0.0000000269
	α_2	0.84999200041440	0.84999999995773	0.84999584987409	0.85000004254726
	$\frac{\lvert\alpha_2-0.85\rvert}{0.85}$	0.000008	**0.0000000000423**	0.00000415	0.0000000425
	α_3	0.95001365433365	0.94999999979106	0.94999945817490	0.94999990594634
	$\frac{\lvert\alpha_3-0.95\rvert}{0.95}$	0.0000137	**0.000000000209**	0.000000542	0.0000000941
	a	0.99999960182130	0.99999999907077	0.99998842441733	1.00000002104131
	$\frac{\lvert a-1.00\rvert}{1.00}$	0.000000398	**0.000000000929**	0.0000116	0.0000000210
	F	0.00110	0.0000000187	0.000136	0.00000441
Worst	α_1	0.89986759049317	0.89999999754274	0.90001851000084	0.90000037757625
	$\frac{\lvert\alpha_1-0.90\rvert}{0.90}$	0.000132	**0.00000000246**	0.0000185	0.000000378
	α_2	0.85024440008037	0.85000000709250	0.84998125561542	0.85000066521416
	$\frac{\lvert\alpha_2-0.85\rvert}{0.85}$	0.000244	**0.00000000709**	0.0000187	0.000000665
	α_3	0.95021128707672	0.94999999845907	0.94996909375371	0.94999907973644
	$\frac{\lvert\alpha_3-0.95\rvert}{0.95}$	0.000211	**0.00000000154**	0.0000309	0.000000920
	a	1.00043167262447	0.99999999647242	0.99993245642478	0.99999938116949
	$\frac{\lvert a-1.00\rvert}{1.00}$	0.000432	**0.00000000353**	0.0000675	0.000000619
	F	0.00253	0.0000000626	0.000377	0.00000982

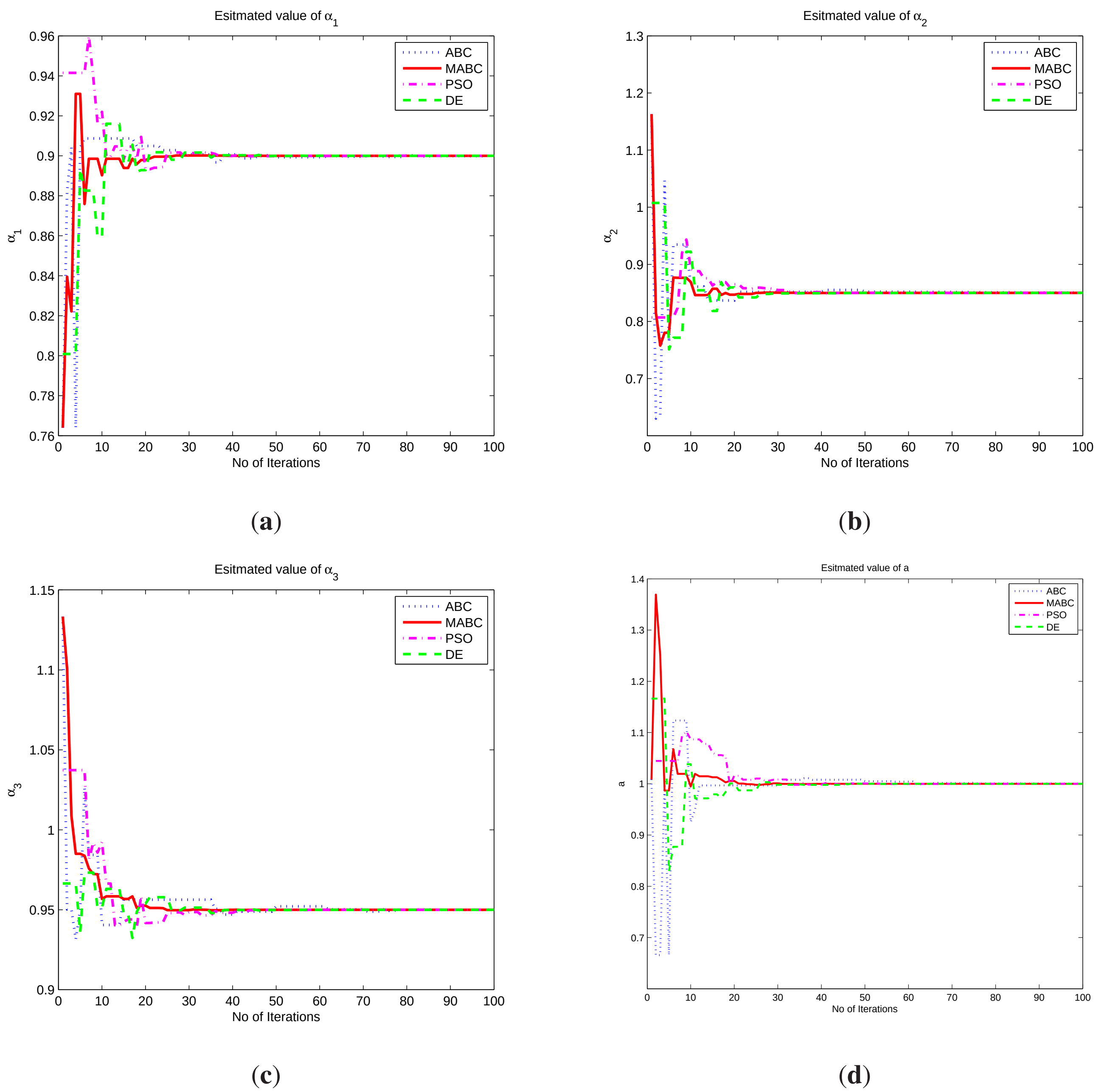


Figure 1. Estimated parameter values with various algorithms on a fractional-order economic system.

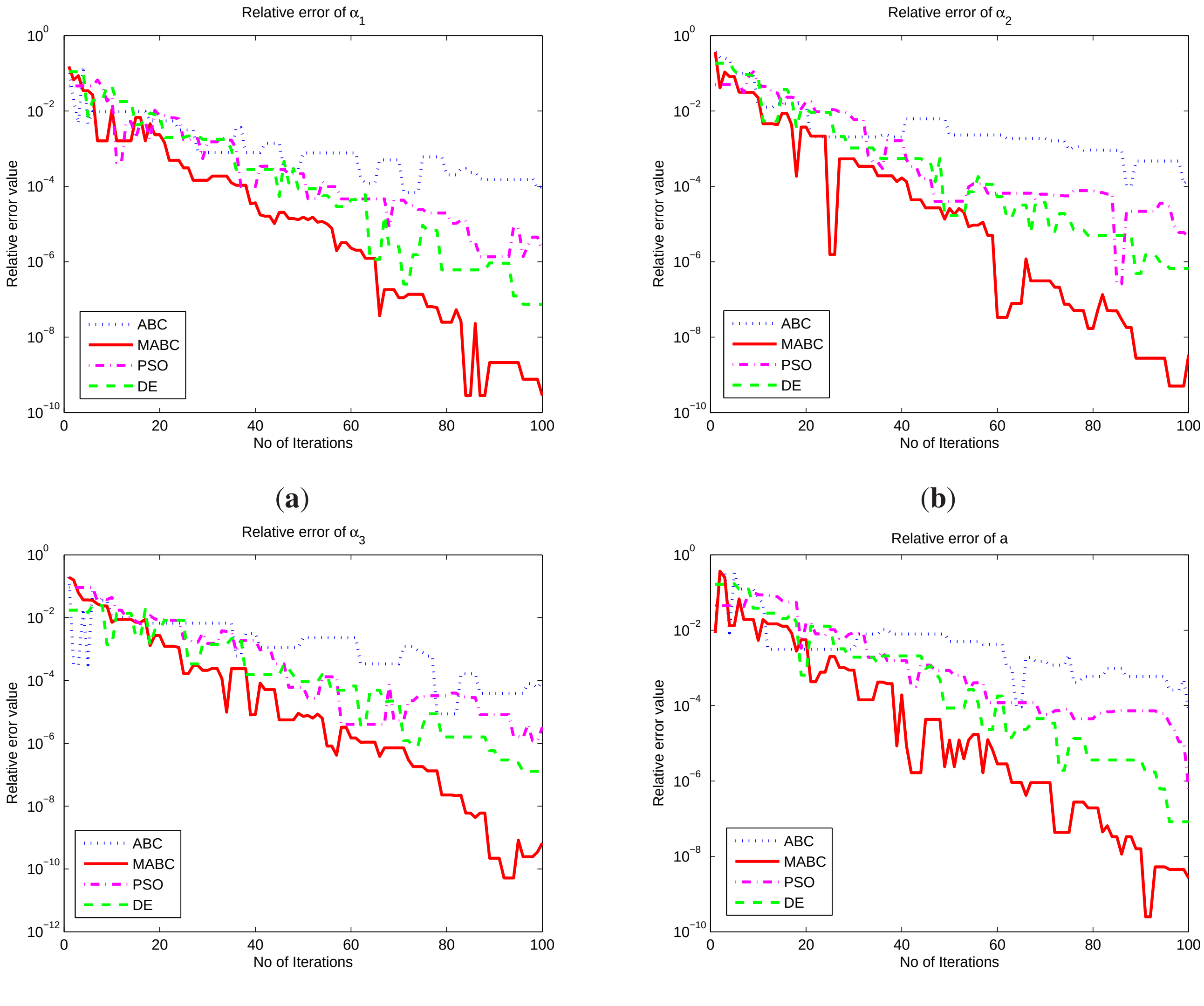


Figure 2. Relative error values with various algorithms on a fractional-order economic system.

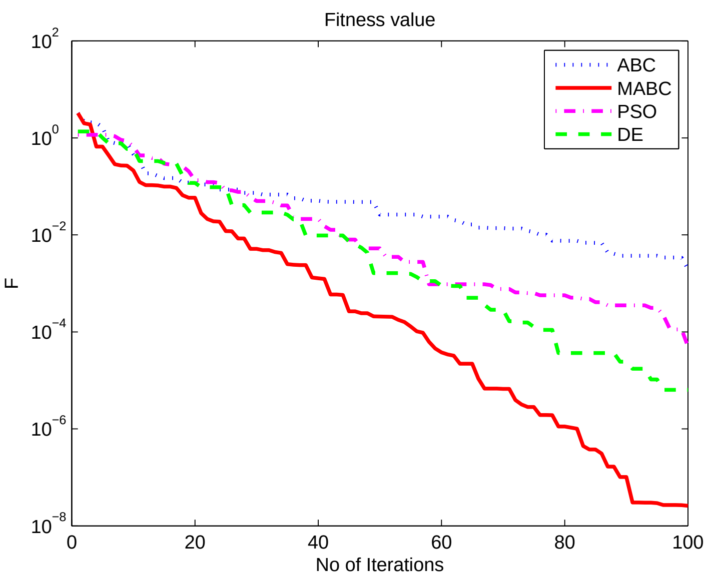


Figure 3. Best fitness values with various algorithms on a fractional-order economic system.

Example 2. *The fractional-order Rössler system [26,28] is described as:*

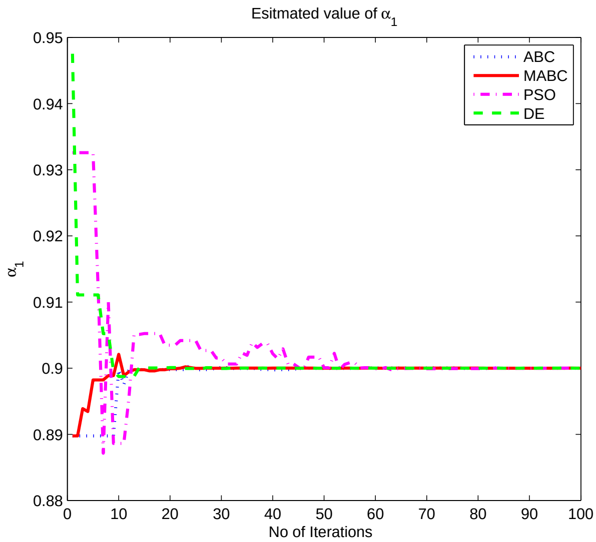
$$\begin{cases} {}_0D_t^{\alpha_1}x(t) = -(y(t)+z(t)), \\ {}_0D_t^{\alpha_2}y(t) = x(t)+ay(t), \\ {}_0D_t^{\alpha_3}z(t) = b+z(t)(x(t)-c), \end{cases} \tag{13}$$

when $(a,b,c) = (0.5, 0.2, 10)$*,* $(\alpha_1,\alpha_2,\alpha_3) = (0.90, 0.85, 0.95)$ *and the initial point is* $(0.5, 1.5, 0.1)$*, System (13) is chaotic. In this example, the fractional orders* $\alpha_1,\alpha_2,\alpha_3$ *are selected as unknown parameters to be estimated. The searching spaces of the unknown parameters are set as* $(\alpha_1,\alpha_2,\alpha_3) \in [0.1, 1.4] \times [0.1, 1.4] \times [0.1, 1.4]$.

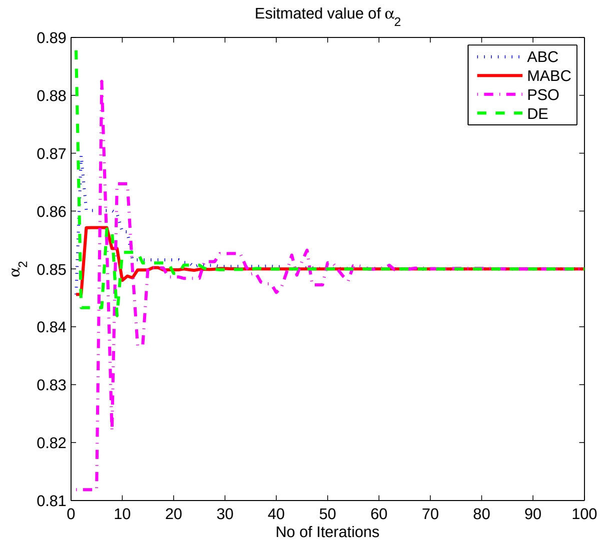
To show the performance of MABC algorithm, the statistical results of the best, the mean and the worst estimated parameters by various algorithms over 15 independent runs are listed in Table 2. From Table 2, it can be seen that the MABC algorithm has more accurate results than those of the standard ABC, PSO and DE algorithms. Figures 4–6 depict the convergence profile of the evolutionary processes of the estimated parameters and the fitness values. From the figures, it can be seen that the MABC algorithm still can converge to the optimal solution more rapidly than the other algorithms.

Table 2. Simulation results of various algorithm for System (13) over 15 independent runs.

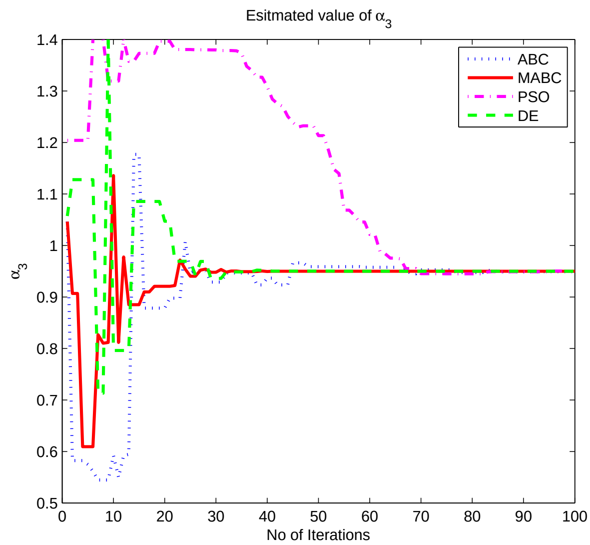
Algorithm		ABC	MABC	PSO	DE
Best	α_1	0.89999988336465	0.90000000000057	0.89999976800825	0.89999999997794
	$\frac{\lvert\alpha_1-0.90\rvert}{0.90}$	0.000000117	**0.000000000000565**	0.000000232	0.0000000000221
	α_2	0.84999973045874	0.84999999999937	0.84999856019088	0.85000000023856
	$\frac{\lvert\alpha_2-0.85\rvert}{0.85}$	0.000000270	**0.000000000000635**	0.00000144	0.000000000239
	α_3	0.94998811621200	0.95000000005566	0.95012518289719	0.95000000987768
	$\frac{\lvert\alpha_3-0.95\rvert}{0.95}$	0.0000119	**0.0000000000557**	0.000125	0.00000000988
	F	0.0000666	0.0000000000405	0.0000828	0.0000000120
Mean	α_1	0.90000227541788	0.89999999997784	0.89999581166110	0.89999999996680
	$\frac{\lvert\alpha_1-0.90\rvert}{0.90}$	0.00000228	**0.0000000000222**	0.00000419	0.0000000000332
	α_2	0.85000037233112	0.85000000004318	0.85001036746114	0.84999999991562
	$\frac{\lvert\alpha_2-0.85\rvert}{0.85}$	0.000000372	**0.0000000000432**	0.0000104	0.0000000000844
	α_3	0.95037664105673	0.95000000056334	0.94957364204493	0.94999999450775
	$\frac{\lvert\alpha_3-0.95\rvert}{0.95}$	0.000377	**0.000000000563**	0.000426	0.00000000549
	F	0.000395	0.00000000328	0.000557	0.0000000234
Worst	α_1	0.90002081204987	0.89999999989872	0.89994777622230	0.90000000092049
	$\frac{\lvert\alpha_1-0.90\rvert}{0.90}$	0.0000208	**0.000000000101**	0.0000522	0.000000000920
	α_2	0.85002033675238	0.85000000049223	0.85012993522134	0.84999998861606
	$\frac{\lvert\alpha_2-0.85\rvert}{0.85}$	0.0000203	**0.000000000492**	0.000130	0.00000000138
	α_3	0.95205165177920	0.95000003505732	0.94352872137521	0.94999989275506
	$\frac{\lvert\alpha_3-0.95\rvert}{0.95}$	0.00205	**0.0000000351**	0.00647	0.000000107
	F	0.000781	0.0000000116	0.00279	0.0000000383



(a)

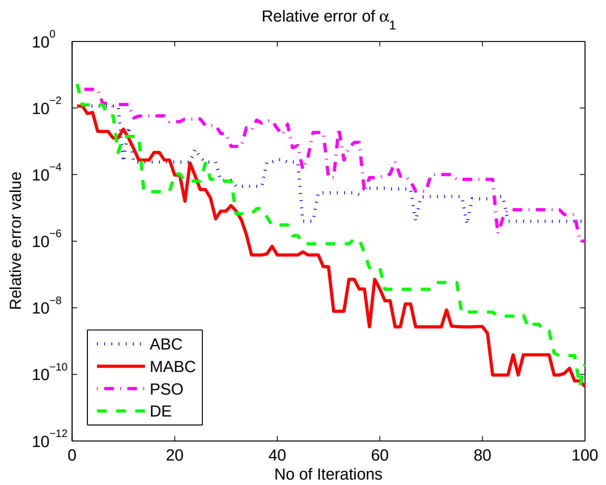


(b)

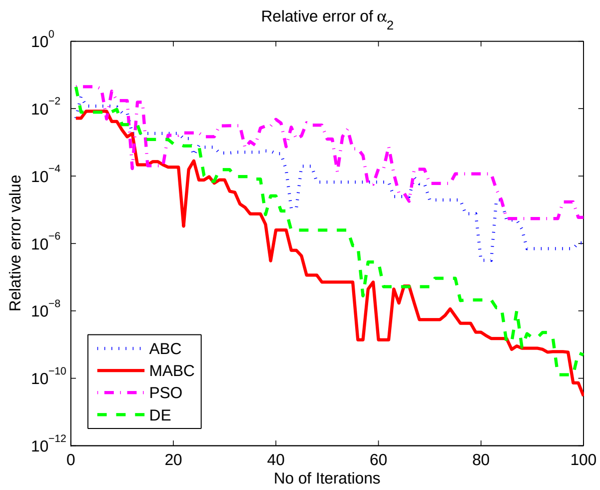


(c)

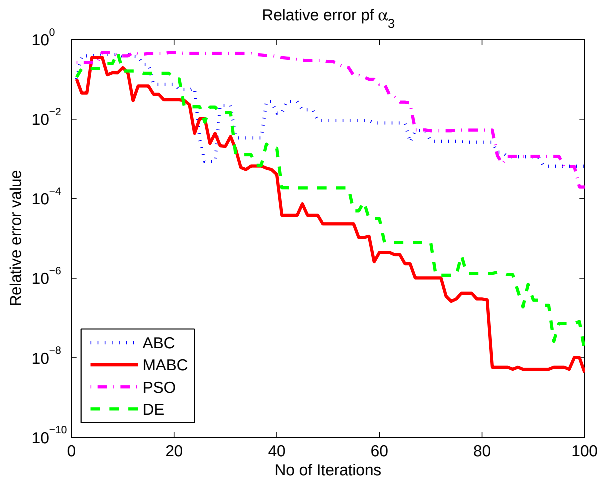
Figure 4. Estimated parameter values with various algorithms on a fractional-order Rössler system.



(a)



(b)



(c)

Figure 5. Relative error values with various algorithms on a fractional-order Rössler system.

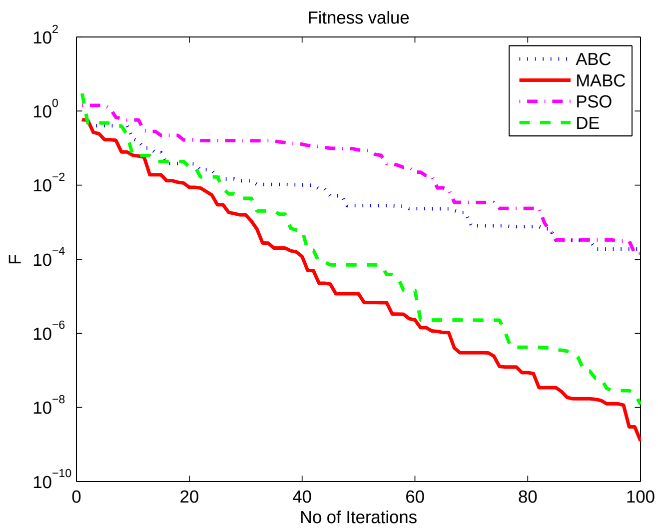


Figure 6. Best fitness values with various algorithms on a fractional-order Rössler system.

Based on the above two examples, the proposed scheme is significantly better than all of the other listed algorithms, and it has a faster convergence speed and higher calculation accuracy in estimating the unknown fractional orders and systematic parameters of fractional-order systems.

6. Conclusions

In this paper, the parameter estimation of fractional-order chaotic systems is converted to an optimization problem from the prospect of optimization. A modified artificial bee colony algorithm is proposed to solve the optimization problem via a functional extrema model. In simulations, the proposed method is employed to identify two typical fractional-order chaotic systems. It is shown that, for the given parameter configurations and maximum number of iterations, the modified ABC algorithm could estimate the unknown fractional orders and parameters of the uncertain fractional-order chaotic systems more rapidly, more accurately and more stably than standard the ABC, PSO and DE algorithms. Additionally, the proposed method can avoid the design of the parameter update law in the synchronization analysis of the fractional-order chaotic systems with unknown parameters. Furthermore, although this paper is mainly concentrated on the parameter estimation problem of the fractional-order chaotic systems, the proposed method is also a useful tool for the study of various numerical optimization problems in physics and other related areas.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (No. 11371049).

Author Contributions

Wei Hu and Yongguang Yu proposed and designed the research; Wei Hu performed the simulations and wrote the paper; Wei Hu, Yongguang Yu and Sha Wang analyzed the simulation results. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Kilbas, A.A.A.; Srivastava, H.M.; Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*; Elsevier: Amsterdam, the Netherlands, 2006.
2. Arena, P.; Caponetto, R.; Fortuna, L.; Porto, D. *Nonlinear Noninteger Order Circuits and Systems—An Introduction*; World Scientific: Singapore, Singapore, 2000.
3. Rivero, M.; Rogosin, S.V.; Tenreiro Machado, J.A.; Trujillo, J.J. Stability of fractional order systems. *Math. Probl. Eng.* **2013**, *2013*, doi:/10.1155/2013/356215.
4. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 27 November–1 December 1995; pp. 1942–1948.

5. Yuan, L.G.; Yang, Q.G. Parameter identification and synchronization of fractional-order chaotic systems. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 305–316.
6. Alfi, A.; Modares, H. System identification and control using adaptive particle swarm optimization. *Appl. Math. Model.* **2011**, *35*, 1210–1221.
7. Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **1997**, *11*, 341–359.
8. Gao, F.; Fei, F.X.; Lee, X.J.; Tong, H.Q.; Deng, Y.F.; Zhao, H.L. Inversion mechanism with functional extrema model for identification incommensurate and hyper fractional chaos via differential evolution. *Expert Syst. Appl.* **2014**, *41*, 1915–1927.
9. Tang, Y.G.; Zhang, X.Y.; Hua, C.C.; Li, L.X.; Yang, Y.X. Parameter identification of commensurate fractional-order chaotic system via differential evolution. *Phys. Lett. A* **2012**, *376*, 457–464.
10. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report-TR06; Erciyes University: Kayseri, Turkey, October 2005.
11. Karaboga, D.; Akay, B. A comparative study of artificail bee colony algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132.
12. Karaboga, D.; Akay, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697.
13. Biswas, S.; Das, S.; Debchoudhury, S.; Kundu, S. Co-evolving bee colonies by forager migration: A multi-swarm based Artificial Bee Colony algorithm for global search space. *Appl. Math. Comput.* **2014**, *232*, 216–234.
14. Li, X.T.; Yin, M.H. Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm. *Nonlinear Dyn.* **2014**, *77*, 61–71.
15. Zhang, W.; Wang, N.; Yang, S.P. Hybrid artificial bee colony algorithm for parameter estimation of proton exchange membrane fuel cell. *Int. J. Hydrog. Energy* **2013**, *38*, 5796–5806
16. Tang, K.S.; Man, K.F.; Kwong, S.; He, Q. Genetic algorithms and their applications. *IEEE Signal Proc. Mag.* **1996**, *13*, 22–37.
17. Price, K.V.; Storn, R.M.; Lampinen, J.A. *Differential Evolution: A Practial Approach to Global Optimization*; Springer: Berlin/Heidelberg, Germany, 2005.
18. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999.
19. Diethelm, K.; Ford, N.J.; Freed, A.D. A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dyn.* **2002**, *29*, 3–22.
20. Gao, W.F.; Liu, S.Y. A modified artificial bee colony algorithm. *Comput. Oper. Res.* **2012**, *39*, 687–697.
21. Zhu, G.P.; Kwong, S. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl. Math. Comput.* **2010**, *217*, 3166–3173.
22. Gao, W.F.; Liu, S.Y.; Huang, L.L. A global best artificial bee colony algorithm for global optimization, *J. Comput. Appl. Math.* **2012**, *236*, 2741–2753.
23. Gao, W.F.; Liu, S.Y.; Huang, L.L. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Trans. Cybern.* **2013**, *43*, 1011–1024.
24. Banharnsakun, A.;Achalakul, T.; Sirinaovakul, B. The best-so-far selection in artificial bee colony algorithm. *Appl. Soft Comput.* **2010**, *11*, 2888–2901.

25. Eberhart, R.C.; Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In Proceedings of the 2000 Congress on Evolutionary Computation, San Diego, CA, USA, 16–19 July 2000; pp. 84–88.
26. Petráš, I. Fractional-order chaotic systems. In *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation*; Spring: Berlin/Heidelberg, Germany, 2011; pp. 103–184.
27. Chen, W.C. Nonlinear dynamics and chaos in a fractional-order financial system. *Chaos Soliton. Fract.* **2008**, *36*, 1305–1314.
28. Li, C.; Chen, G.R. Chaos and hyperchaos in the fractional-order Rössler equations. *Physica A* **2004**, *341*, 55–61.