

Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm

Xiangtao Li · Minghao Yin

Received: 10 November 2013 / Accepted: 25 January 2014 / Published online: 22 February 2014
© Springer Science+Business Media Dordrecht 2014

Abstract This paper is concerned with the parameter estimation of nonlinear chaotic system, which could be essentially formulated as a multi-dimensional optimization problem. In this paper, a hybrid algorithm by combining differential evolution with artificial bee colony is implemented to solve parameter estimation for chaotic systems. Hybrid algorithm combines the exploration of differential evolution with the exploitation of the artificial bee colony effectively. Experiments have been conducted on Lorenz system and Chen system. The proposed algorithm is applied to estimate the parameters of two chaotic systems. Simulation results and comparisons demonstrate that the proposed algorithm is better or at least comparable to differential evolution, artificial bee colony, particle swarm optimization, and genetic algorithm from literature when considering the quality of the solutions obtained.

Keywords Differential evolution algorithm · Artificial bee colony · Exploration · Exploitation · Chaotic system · Parameter estimation

1 Introduction

As a quintessence of nonlinear systems, chaos is a bounded unstable dynamic behavior that exhibits sensitive dependence on the initial conditions and includes infinite unstable periodic motions [1]. Chaos has been applied in many academic and engineering fields, such as communication, economic system, and optimization. During the last decade, we have viewed significant progresses on deal with the parameters of chaotic systems. Moreover, in the real world, the parameters may be difficult to determine due to the complexity of chaotic systems. Therefore, the parameter estimation for chaotic system has become an important issue of nonlinear science. So far, different kinds of classical techniques have been advanced to handle these problems [2–7].

Among them, meta-heuristic-based method, such as genetic algorithm (GA), particle swarm optimization algorithm (PSO), and differential evolution algorithm (DE) [2, 5, 6, 8], may be one of the most popular methods by defining the problem as a multi-dimensional optimization problem. Parameter estimation for chaotic systems using these heuristic algorithms is reported in some research. The paper [9] proposes a variant of particle swarm optimization, called Drift particle swarm optimization, and applies it in estimating the unknown parameters of chaotic dynamic systems. The principle and procedure of DPSO are presented, and the algorithm is used to identify the Lorenz system and the Chen system. Experimental results show that

X. Li · M. Yin (✉)
School of Computer Science and Information Technology,
Northeast Normal University, Changchun 130117, China
e-mail: Minghao.Yin1@gmail.com

the DPSO can identify the parameters of the systems accurately and effectively, and may be a promising tool for chaotic system. The paper [3] presents that a PSO approach is implemented to solve parameter estimation for chaotic systems. In this paper, parameter estimation for chaotic system is formulated as a multi-dimensional optimization problem. The paper [9] combines biogeography-based optimization with differential evolution and simplex search to develop an effective hybrid algorithm for parameter estimation for chaotic system that is formulated as a multi-dimensional optimization problem. Numerical simulation based on two chaotic systems and comparisons with some existing methods demonstrate the effectiveness of the proposed algorithm. The paper [10] presents an effective hybrid quantum-inspired evolutionary algorithm with differential evolution to estimate the parameters of the Lorenz system. Numerical simulation and comparisons show that the proposed algorithm is effective and efficient.

Differential evolution algorithm [11] and artificial bee colony algorithm (ABC) [12] are two novel meta-heuristic algorithms introduced recently and have gained significant attention in the research literatures [13, 14]. DE is a simple yet powerful population-based, direct search algorithm with the generation and test feature for global optimization problems. The basic idea of DE is to create new candidate solutions by combining the parent individual and several other individuals of the same population, and a candidate solution replaces the parent only if it has better fitness. Artificial bee colony algorithm is a population-based heuristic evolutionary algorithm inspired by the intelligent foraging behavior of the honeybee swarm.

On the other hand, although meta-heuristic methods have been proved to have superior features to other traditional methods, they often suffer some limitations. In some cases, they may easily fall into the local minimum or converge too slowly. For example, for DE algorithm, it is good at exploring the search space and locating the region of global minimum but it is slow at exploitation of the solution [11]. Recently, researchers have found that a skilled combination of two meta-heuristic techniques can improve the performance of the algorithms obviously when dealing with real-world and large-scale problems [15]. Several hybrid heuristic-based optimization methods have been investigated in the past few years [16, 17].

However, this field of study is still in its early days; a large number of future researches is necessary in order to develop hybrid algorithm for optimization problems.

In this paper, parameter estimation for chaotic system is formulated as a multi-dimensional optimization problem; and a hybrid differential evolution algorithm and artificial bee colony algorithm are implemented to solve this problem. Hybrid bee operator is proposed to combine the DE operator with the ABC operation. It can balance the exploration and exploitation of the algorithm. To the best of our knowledge, this is the first paper to apply hybrid differential evolution algorithm and artificial bee colony to estimate the parameter of chaotic system. Experiments have been conducted on the Lorenz system and the Chen system. Simulation results and comparisons demonstrate the effectiveness of the proposed algorithm compared with other algorithms.

The rest of this paper is organized as follows: in Sect. 2 we will give the problem formulation. Section 3 describes the differential evolution algorithm. Section 4 describes the artificial bee colony algorithm. In Sect. 5, we introduce the hybrid algorithm. Benchmark problems and the corresponding experimental results are given in Sect. 6. In the last section we conclude this paper and point out some future research directions.

2 Problem formulation

This section describes the chaotic system. Let

$$\dot{X} = F(X, X_0, \theta_0) \quad (1)$$

be a continuous nonlinear chaotic system, where the state vector of the chaotic system is $X = (x_1, x_2, \dots, x_N)' \in R^n$, \dot{X} is the derivative of X , and X_0 denotes the initial state. $\theta_0 = (\theta_{10}, \theta_{20}, \dots, \theta_{d0})'$ are the original parameters.

Suppose the structure of system (1) is known, then the estimated system can be written as

$$\dot{\tilde{X}} = F(X, X_0, \tilde{\theta}), \quad (2)$$

where $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N)' \in R^n$ denotes the state vector, and $\tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2, \dots, \tilde{\theta}_d)'$ is a set of estimated parameters.

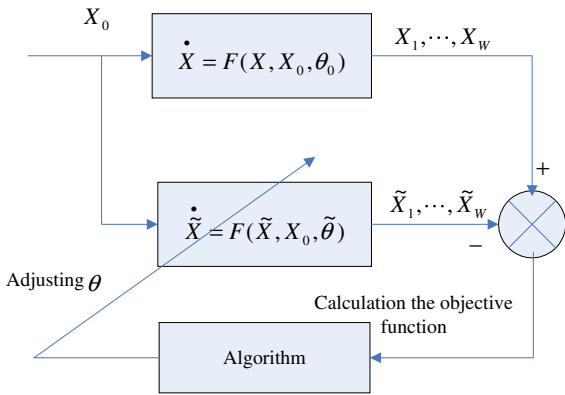


Fig. 1 The principle of parameter estimation for a chaotic system

Based on the measurable state vector $X = (x_1, x_2, \dots, x_N)' \in R^n$, we define the following objective function or fitness function

$$f(\tilde{\theta}_i^n) = \sum_{t=0}^W [(x_1(t) - x_{i,1}^n(t))^2 + \dots + (x_N(t) - x_{i,N}^n(t))^2], \tag{3}$$

where $t = 0, 1, \dots, W$. The goal of estimating the parameters of chaotic system (2) is to find out the suitable value of $\tilde{\theta}_i^n$ such that fitness function (3) is globally minimized.

Obviously, the parameter estimation for chaotic system is a multi-dimensional continuous optimization problem, where the decision vector is θ and the optimization is to minimize $f(\tilde{\theta}_i^n)$. Figure 1 shows the principle of parameter estimation for chaotic system in a sense of optimization.

The chaotic systems are not easy to estimate because of the unstable dynamic of the chaotic system. Moreover, due to multiple variables in the problem and multiple local search optima in the objective functions, traditional optimization can easily trap in local optima. Therefore, we aim to solve this problem by proposing a hybrid differential evolution and artificial bee colony algorithm in this paper.

3 Differential evolution algorithm

Differential evolution (DE) is a simple optimization method that has parallel, direct search, easy to use, good convergence, and fast implementation properties. Similar to other evolutionary algorithms particularly genetic algorithm, DE uses some evolutionary operators like selection recombination and mutation operators. Different from genetic algorithm, DE uses distance and direction information from current population to guide the search process. The crucial idea behind DE is a scheme for generating trial parameter vectors. If the trail vector yields a smaller value than a predetermined population member, the newly trail vector will be accepted and be compared in the following generation. Different kinds of strategies of DE have been proposed based on the target vector selected and the number of difference vectors used. In this paper, we use the DE/rand/2/bin mutation strategy as the final mutation strategy. The detail description of the differential evolution algorithm can be found in the paper [18].

4 Artificial bee colony

Artificial bee colony introduced by Karaboga in 2005 is the recent swarm-based evolutionary computation technique and models the foraging behavior of honey bees. In this algorithm, the model of the ABC algorithm consists of three groups of bees: employed bees, onlooker bees, and scout bees. For each food source, there is only one employed bee. In other words, the number of bees is equal to the number of food sources. Employed bees are responsible for exploiting the sources explored before, and sharing their information with onlookers within the hive. After that, the onlookers will select one of the food sources within the neighborhood of the food source. An employed bee becomes a scout if the food source is abandoned, and then starts to search a new food source randomly [9]. The main steps of the ABC algorithm simulating these behaviors are listed below:

procedure *Artificial bee colony Algorithm*

begin

Initialize

Repeat

Step1: Move the employed bees onto their food sources and determine their nectar amounts.

Step2: Move the onlookers onto the food sources and determine their nectar amounts.

Step3: Move the scouts for searching new food sources.

Step4: Memorize the best food source found so far.

UNTIL (requirement are met)

end.

As we can see in the algorithm, each cycle of the search consists of three steps: moving the employed and onlooker bees onto the food sources; calculating their nectar amounts, respectively; and then determining the scout bees and moving them randomly onto the possible food source. Here, a food source stands for a potential solution of the problem to be optimized. The ABC algorithm is an iterative algorithm. The detail description of the artificial bee colony can be found in the paper [19].

5 Our approach: DE/ABC

As mentioned above, DE can explore the search space better and find the region of global minimum. However, it is slow at exploitation of the optimal solution. In ABC, employed bees are responsible for exploiting the nectar sources explored before and giving information to the waiting bee in the hive about the quality of the food source sites which they are exploiting. Onlooker bees wait in the hive and decide on a food source to exploit based on the information shared by the employed bees. The algorithm has a good exploitation for global optimization. In order to achieve better results on these benchmark functions, DE/ABC is proposed to solve these problems. This algorithm can take advantage of the exploration capability of the DE algorithm and the stochastic exploitation ability of the ABC [11, 12]. The main framework of the proposed algorithm can be summarized as follows: first,

a hybrid bee operator is proposed; which combines the exploration of DE and the exploitation of ABC effectively. Second, we introduced a novel boundary constraint; which is reflected back from the bound by the amount of violation. Finally, we propose a hybrid algorithm combining differential evolution with artificial bee colony for the parameter estimation of chaotic system.

5.1 Hybrid bee operator

The main operator of DE/ABC is the hybrid bee operator, which hybridizes the DE operator with the ABC operator. Crossover operation of the DE algorithm is applied to each pair of the target vector $x_{j,i,G}$ and its corresponding mutate vector $v_{j,i,G}$ to generate a trail vector. In the standard DE algorithm, DE employs the binomial crossover defined as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & (\text{rand}_j[0, 1] \leq \text{CR}) \text{ or } (j = j_{\text{rand}}) \\ x_{j,i,G}, & \text{otherwise} \end{cases} \quad (4)$$

Crossover operator can find the globally optimal region. However, it cannot converge rapidly to the globally optimal solution. Employed bees fly onto the source which they are exploiting. In order to solve this problem, we tackle by integrating an employed bee operator of ABC to maintain the diversity and obtain good solution rapidly at the same time.

procedure hybrid bee operator of DE/ABC

```

begin
  for  $i=1$  to NP
    select randomly  $a \neq b \neq c \neq d \neq e \neq f \neq i$ 
    for  $j=1$  to  $D$ 
       $j_{rand} = \lfloor rand(0,1) * D \rfloor$ 
      If  $rand(0,1) \leq CR$  or  $j = j_{rand}$  then
         $u_{i,j} = x_{a,j} + F \times (x_{b,j} - x_{c,j}) + F \times (x_{d,j} - x_{e,j})$ 
      else
        If  $rand < 0.2$ 
           $u_{i,j} = x_{i,j} + (x_{i,j} - x_{f,j}) * \varphi_{i,j}$ 
        end if
      end if
    end for
  end for
end.

```

The crucial idea behind DE is a scheme for generating trial parameter vectors according to the manipulation of target vector and difference vector. From the algorithm, we can find that the proposed hybrid bee operation is based on the main update operator of ABC. The core idea of the proposed hybrid bee colony is based on two considerations. On the one hand, employed bee colony exploited the nectar sources explored before—giving the information to the waiting bees in the hive about the quality of the food source sites which they are exploiting and onlooker bees waiting in the hive watch the dances advertising the profitable sources and choose a source site depending on the frequency of a dance proportional to the quality of the source. On the other hand, the mutation operator of the DE is able to explore the new space. From the analysis, it can be seen that the hybrid bee colony can balance the exploration and the exploitation effectively.

5.2 Boundary constraints

The DE/ABC algorithm assumes that the whole population should be in an isolated and finite space. During the searching process, if there are some individuals that will move out to the bounds of the space, the original algorithm stops them on the boundary. In other words, the individual will be assigned a boundary value. The disadvantage is that if there are too many individuals on the boundary, and especially when there exists some local minimum on the boundary, the algorithm will lose its population diversity to some extent. In order to tackle this problem, we propose the following repair rule:

$$x_i = \begin{cases} l_i + \text{mod}((l_i - x_i), (u_i - l_i)) \\ u_i - \text{mod}((x_i - u_i), (u_i - l_i)) \end{cases} \tag{5}$$

5.3 Proposed DE/ABC algorithm

In this section, we introduce the newly proposed DE/ABC. The DE/ABC algorithm incorporates the ABC operator into DE. The crossover can choose the individual by the CR. The algorithm begins with a randomly initiated population which generates NP*D matrix with uniform probability distribution random values. We can generate the j th component of the i th vector as

$$x_{j,i,0} = x_{j,\min} + \text{rand}_{i,j}[0, 1] \cdot (x_{j,\max} - x_{j,\min}), \tag{6}$$

where $\text{rand}_{i,j}[0, 1]$ is a uniformly distribution random number between 0 and 1. $i = 1, \dots, NP$ and $j = 1, \dots, D$. $x_{j,\max}, x_{j,\min}$ is the upper bound and lower bound of the j th column, respectively. Then, the algorithm uses a hybrid bee operation in which the mutated individual $v_{i,G}$ is mated with the ABC operator, DE operator, and generate the offspring or trail individual $u_{i,G}$. The genes of $u_{i,G}$ are inherited from the $v_{i,G}$ of DE and the ABC operators. Thus, the scout bee colony can update the abandoned population. In addition, the DE/ABC has a very simple structure and thus is easy to implement. Moreover, this algorithm combines the exploration capability of the DE algorithm and the stochastic exploitation of the ABC. This method can overcome the lack of the exploitation of the DE algorithm. The DE/ABC was described as follows:

procedure Algorithm description of DE/ABC**begin****Step 1:** Set the generation counter $G=0$; and randomly initialize a population of NP individuals X_i . Initialize the parameter F , CR, limit.**Step 2:** Evaluate the fitness for each individual in P .**Step 3:** while stopping criteria is not satisfied do

%% hybrid employed bee colony

 for $i=1$ to NP select randomly $a \neq b \neq c \neq d \neq e \neq f \neq i$ for $j=1$ to D $j_{rand} = \lfloor rand(0,1) * D \rfloor$ If $rand(0,1) \leq CR$ or $j == j_{rand}$ then $u_{i,j} = x_{a,j} + F \times (x_{b,j} - x_{c,j}) + F \times (x_{d,j} - x_{e,j})$

else

 If $rand < 0.2$ $u_{i,j} = x_{i,j} + (x_{i,j} - x_{f,j}) * \varphi_{i,j}$

end if

end if

end for

end for

%% hybrid Onlooker bee colony

 $i=1$; $t=0$; while $t < NP$ if $rand < prob(i)$ $t=t+1$; select randomly $a \neq b \neq c \neq d \neq e \neq f \neq i$ for $j=1$ to D $j_{rand} = \lfloor rand(0,1) * D \rfloor$ If $rand(0,1) \leq CR$ or $j == j_{rand}$ then $u_{i,j} = x_{a,j} + F \times (x_{b,j} - x_{c,j}) + F \times (x_{d,j} - x_{e,j})$

else

 If $rand < 0.2$ $u_{i,j} = x_{i,j} + (x_{i,j} - x_{f,j}) * \varphi_{i,j}$

end if

end if

end for

end if

 $i=i+1$; if $i == NP+1$ $i=1$;

end if

end while

%% Scout bee colony

 If $max(trial_i) > limit$ Replace x_i with a new randomly produced solution by $x_{ij} = l_j + (u_j - l_j) \times r$

End if

Memorize the best solution achieved so far

Step 4: end while**end**

6 Simulation results

To evaluate the performance of our algorithm, we apply it to two chaotic systems as the standard benchmarks. The Lorenz chaotic system and the Chen chaotic system described below were chosen to test the performance of the algorithm. So far, these problems have been widely used as benchmarks for studies with different methods by many researchers. Each algorithm ran 30 times on each system. The parameters of the algorithm were configured as follows. After periods of transient process, a state vector is chosen as the initial state X_0 for parameter estimated systems as shown in Fig. 1. The successive W states ($W = 30$) of both the estimated system and the original system are used to calculate the fitness. For the DE/ABC, the step size F is 0.5. The crossover rate CR is 0.9. The value of “limit” is 100. For the differential evolution algorithm, the step size F is 0.5. The crossover rate CR is 0.9. For ABC algorithm, the value of “limit” is 100. For PSO algorithm, the inertia weight is 0.729 and the acceleration coefficients were set to be 1.49. For GA, the crossover rate is 0.8 and the mutation rate is increasing linearly from 0.001 to 0.6 during the evolution. In order to perform fair comparison, the maximum generation number is 100 for the Lorenz system and 200 for the Chen system. The population size of each algorithm is 30. All algorithms are coded in MATLAB 7.0, and experiments are made on a Pentium 3.0GHz Processor with 1.0GB of memory.

6.1 Simulation the Lorenz system

In the first example, the well-known Lorenz system [20] is employed as an example in this paper. The general expression of the chaotic system can be described as follows:

$$\begin{cases} \dot{x}_1 = \theta_1(x_2 - x_1) \\ \dot{x}_2 = (\theta_2 - x_3)x_1 - x_2, \\ \dot{x}_3 = x_1x_2 - \theta_3x_3 \end{cases}$$

where $x_1, x_2,$ and x_3 are the state variables; $\theta_1, \theta_2,$ and θ_3 are unknown positive constant parameters. The original parameters are $\theta_1 = 10, \theta_2 = 28,$ and $\theta_3 = 8/3$. In order to simulate, we let the parameters of the Lorenz system be $\theta_1 = 10, \theta_2 = 28,$ and $\theta_3 = 8/3$.

In order to simulate this system, the successive state W is 30 and each algorithm ran 30 times with each single runs 100 iterations. The statistical results of the best fitness value, the mean value, the standard deviation, and the identified parameters of the Lorenz system are shown in Table 1. From Table 1, it can be seen that the best fitness values obtained by DE/ABC can performance better than DE, ABC, PSO, and GA. The mean of the identified parameters by DE/ABC and DE converges more quickly than those identified by

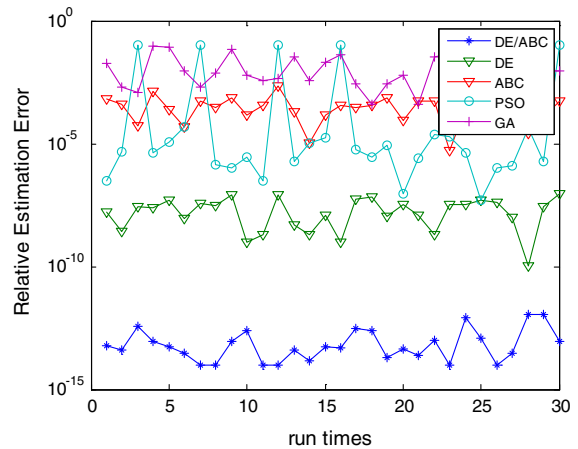


Fig. 2 The relative estimation error of θ_1 is defined by $|\hat{\theta}_1 - 10|/10$

Table 1 Statistical results from the four algorithms on the Lorenz system

Algorithm	Means of the best fitness	SD of the best fitness	Obtained mean value and best value (in brackets) of the identified parameters		
			θ_1	θ_2	θ_3
DE/ABC	9.265e-23	1.2568e-23	10.0000 (10.0000)	28.0000 (28.0000)	2.6667 (2.6667)
ABC	6.6288e-005	1.3950e-004	9.99796 (10.0000)	27.9998 (28.0000)	2.6665 (2.6667)
DE	2.56E-13	3.28E-13	10.0000 (10.0000)	28.0000 (28.0000)	2.6667 (2.6667)
PSO	0.11788	0.268094	10.1667 (9.9999)	28.0105 (27.9999)	2.6684 (2.6666)
GA	1.331401	2.783125	9.9824 (10.0274)	28.0175 28.0058	2.6824 (2.6692)

Bold values indicate that our algorithm is better than other algorithms

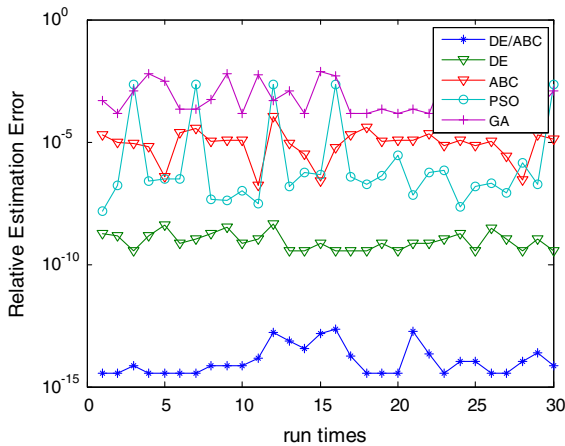


Fig. 3 The relative estimation error of θ_2 is defined by $|\tilde{\theta}_2 - 28|/28$

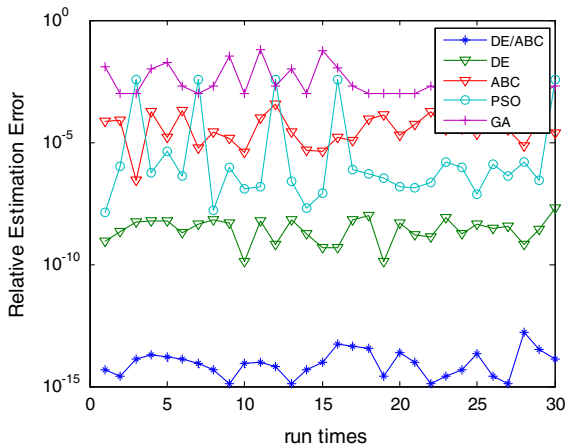


Fig. 4 The relative estimation error of θ_3 is defined by $|\tilde{\theta}_3 - 8/3|/(8/3)$

ABC, PSO, and GA. Figures 2, 3, and 4 show the relative estimation errors from 30 runs of each algorithm with each single run executing 100 iterations. For the relative estimation errors, the three parameters can be defined by $|\tilde{\theta}_1 - 10|/10$, $|\tilde{\theta}_2 - 28|/28$, and $|\tilde{\theta}_3 - 8/3|/(8/3)$, respectively; where $\tilde{\theta}_1$, $\tilde{\theta}_2$, and $\tilde{\theta}_3$ are identification values of the parameters. As can be seen in these figures, we can find that the DE/ABC algorithm is the winner and the algorithm can provide the better estimation errors among all the algorithms. Figure 5 shows that DE/ABC can converge to the optimal solution more rapidly than other algorithms. Figures 6, 7, and 8 show the convergence process of these three parameters, where each algorithm in the single run gen-

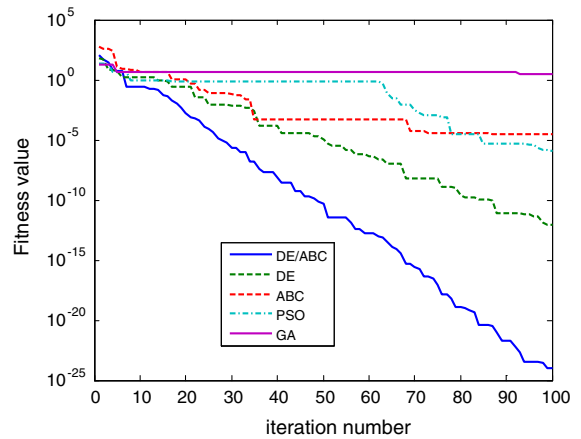


Fig. 5 The convergence process of fitness value averaged over 30 runs for the Lorenz system

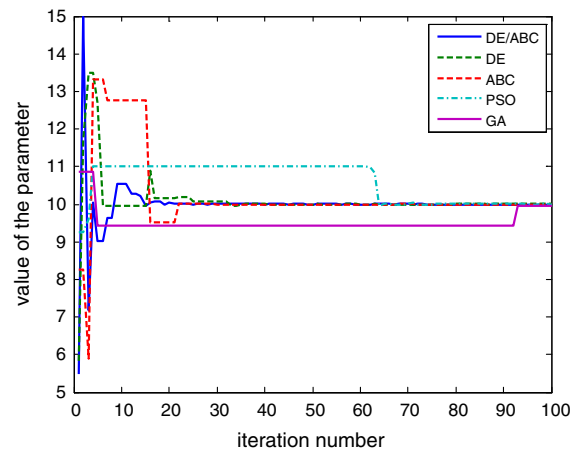


Fig. 6 The convergence process of θ_1 in the single run generating the best solution

erates the best solution. In general, the performance of DE/ABC is highly competitive with DE, ABC, PSO, and GA for the well-known Lorenz system.

6.2 Simulation the Chen system

The second example system is the Chen system [21, 22]. The model of this system can be described as follows:

$$\begin{cases} \dot{x}_1 = \theta_1(x_2 - x_1) \\ \dot{x}_2 = (\theta_3 - \theta_1)x_1 - \theta_3x_2 - x_1x_3, \\ \dot{x}_3 = x_1x_2 - \theta_2x_3 \end{cases}$$

where x_1 , x_2 , and x_3 are the state variables; θ_1 , θ_2 , and θ_3 are unknown positive constant parameters. The orig-

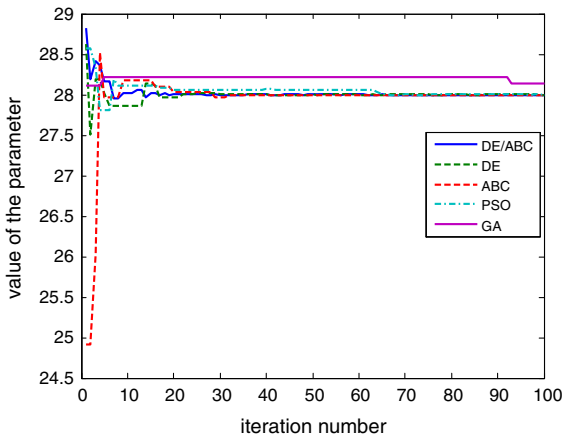


Fig. 7 The convergence process of θ_2 in the single run generating the best solution

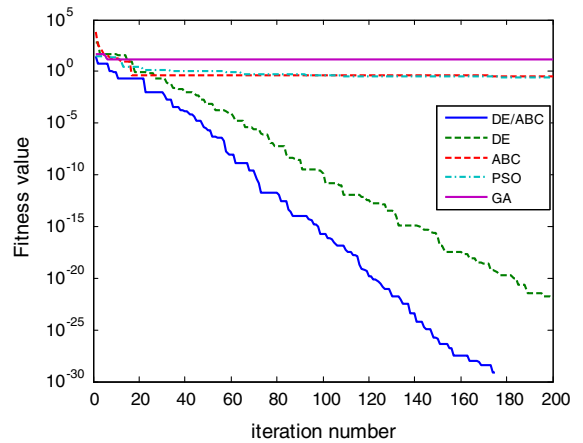


Fig. 9 The convergence process of fitness value averaged over 30 runs for the Chen system

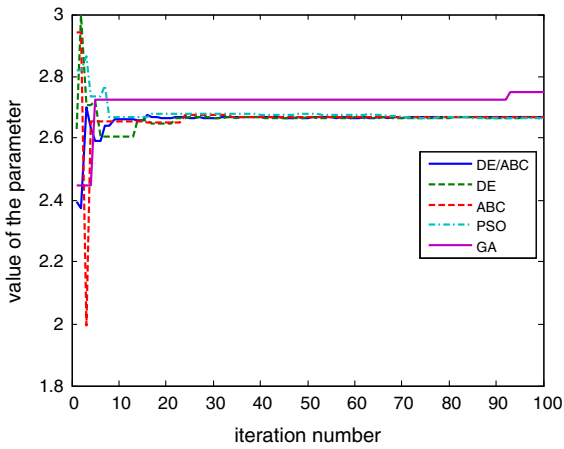


Fig. 8 The convergence process of θ_3 in the single run generating the best solution

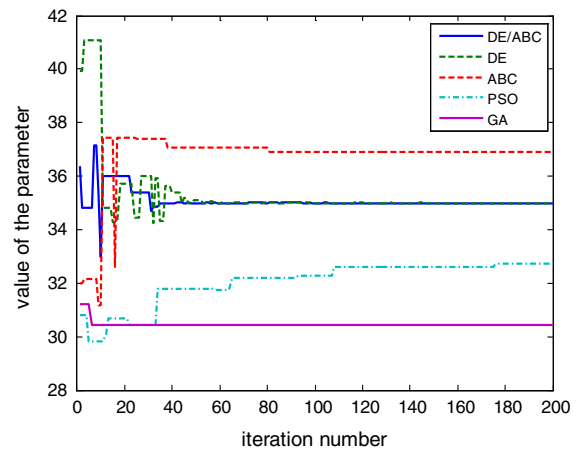


Fig. 10 The convergence process of θ_1 in the single run generating the best solution

inal parameters are $\theta_1 = 35, \theta_2 = 3$, and $\theta_3 = 28$. In order to simulate this system, we let the parameters of the Lorenz system be $\theta_1 = 35, \theta_2 = 3$, and $\theta_3 = 28$.

In order to simulate this system, the successive state W is 30 and each algorithm ran 30 times with each single runs 200 iterations. It seems that the parameters of the Chen system are more difficult to identify

Table 2 Statistical results from the four algorithms on the Chen system

Algorithm	Means of the best fitness	SD of the best fitness	Obtained mean value and best value (in brackets) of the identified parameters		
			θ_1	θ_2	θ_3
DE/ABC	6.5081e-31	1.2368e-31	35.0000 (35.0000)	3.0000 (3.0000)	28.0000 (28.0000)
ABC	0.525804	0.446762	35.23983 (36.01104)	2.997600 (2.996633)	28.119097 (28.491249)
DE	2.31E-22	5.28E-22	35.0000 (35.0000)	3.0000 (3.0000)	28.0000 (28.0000)
PSO	8.461767	24.18407	38.1409 (34.4876)	3.0556 (2.9968)	29.4815 (27.7544)
GA	4.769291	21.53209	35.2407 35.2199	2.9884 (3.0196)	28.1173 (28.0935)

Bold values indicate that our algorithm is better than other algorithms

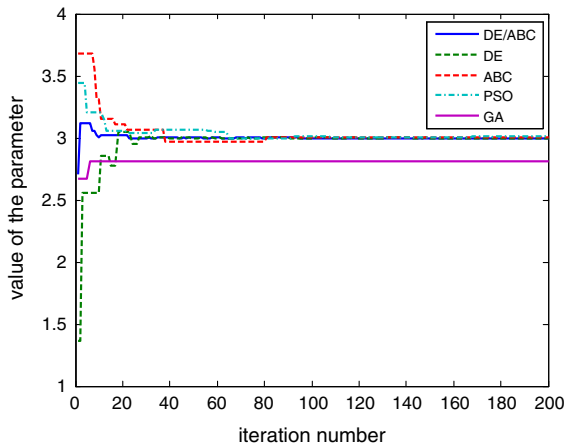


Fig. 11 The convergence process of θ_2 in the single run generating the best solution

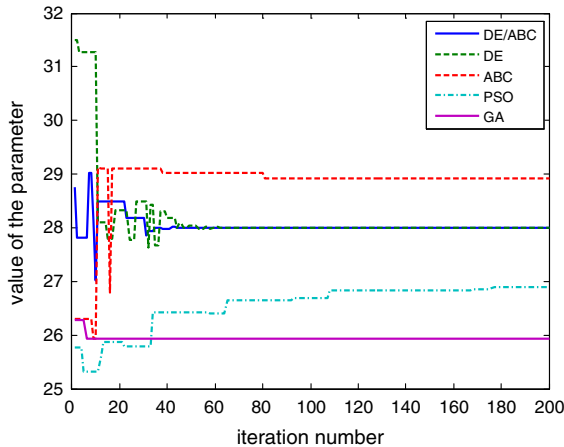


Fig. 12 The convergence process of θ_3 in the single run generating the best solution

than the Lorenz system. The statistical results of the best fitness value, the mean fitness value, the standard deviation, and the identified parameters of the Chen system are listed in Table 2. From Table 2, it can be seen that the best fitness values obtained by DE/ABC algorithm is much better than DE, ABC, PSO, and GA. Figure 9 shows that DE/ABC can converge to the optimal solution more rapidly than the other algorithms. Figures 10, 11, and 12 show the convergence process of these three parameters with each algorithm in the single run generating the best solution.

7 Conclusions

In this paper, the parameter estimation for chaotic system was formulated as a multi-dimensional optimization problem. A novel evolutionary algorithm, hybrid differential evolution and artificial bee colony was applied to solve the parameter estimation for chaotic system. A new hybrid bee operator is proposed to generate the promising solutions. It can balance the exploration and exploitation of the algorithm. In addition, the DE/ABC has a very simple structure and thus is easy to implement. To verify the performance of DE/ABC, the algorithm has been employed for estimation of the parameters of two chaotic dynamical systems. By a comparative study, it is shown that the proposed method is well suited to solve a complex problem. The experimental results show that DE/ABC can identify the parameters of the Lorenz system and the Chen system more accurately, more rapidly, and more stably than DE, ABC, PSO, and GA. The future work is to apply the proposed hybrid algorithm to other chaotic systems and make the DE/ABC a promising tool for various numerical optimization problems in physics.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant Nos. 61370052, 61370156, and 11226275; Natural Science Foundation of Jilin Province under Grant No. 201215006; and Program for New Century Excellent Talents in University under Grant NCET-13-0724. We thank Quan Liu for correcting the English in this manuscript.

References

- Chen, G., Dong, X.: From Chaos to Order: Methodologies, Perspectives, and Applications. World Scientific, Singapore (1998)
- Dai, D., Ma, X.K., Li, F.C., You, Y.: An approach of parameter estimation for a chaotic system based on genetic algorithm. *Acta Phys. Sinica* **11**, 2459–2462 (2002)
- He, Q., Wang, L., Liu, B.: Parameter estimation for chaotic systems by particle swarm optimization. *Chaos Solitons Fractals* **34**, 654–661 (2007)
- Inés, P.M., Joaquín, M.: An approximate gradient-descent method for joint parameter estimation and synchronization of coupled chaotic systems. *Phys. Lett. A* **351**, 262–267 (2006)
- Li, L.X., Yang, Y.X., Peng, H.P., Wang, X.D.: Parameters identification of chaotic systems via chaotic ant swarm. *Chaos Solitons Fractals* **28**, 1204–1211 (2006)
- Peng, B., Liu, B., Zhang, F.Y., Wang, L.: Differential evolution algorithm based parameter estimation for chaotic systems. *Chaos Solitons Fractals* **39**, 2110–2118 (2009)

7. Wang, L., Tang, F., Wu, H.: Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. *Appl. Math. Comput.* **171**, 1141–1156 (2005)
8. Sun, J., Zhao, J., Wu, X.J., Fang, W., Cai, Y.J., Xu, W.B.: Parameter estimation for chaotic systems with a Drift Particle Swarm Optimization method. *Phys. Lett. A* **374**, 2816–2822 (2010)
9. Xu, Y., Wang, L.: An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems. *Expert Syst. Appl.* **30**, 15103–15109 (2011)
10. Wang, L., Li, L.P.: An effective hybrid quantum-inspired evolutionary algorithm for parameter estimation of chaotic systems. *Expert Syst. Appl.* **37**, 1279–1285 (2010)
11. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**, 341–359 (1997)
12. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **8**, 687–697 (2008)
13. Chahkandi, V., Yaghoobi, M., Veisi, G.: CABC-CSA: a new chaotic hybrid algorithm for solving optimization problems. *Nonlinear Dyn.* **73**, 1–10 (2013)
14. Chen, C.H.: Compensatory neural fuzzy networks with rule-based cooperative differential evolution for nonlinear system control. *Nonlinear Dyn.* **75**, 1–12 (2013)
15. Sun, J., Zhang, Q., Tsang, E.: DE/EDA: a new evolutionary algorithm for global optimization. *Inf. Sci.* **169**, 249–262 (2004)
16. Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Inf. Sci.* **177**, 3918–3937 (2007)
17. Choi, D.H.: Cooperative mutation based evolutionary programming for continuous function optimization. *Oper. Res. Lett.* **30**(3), 195–201 (2002)
18. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evolut. Comput.* **15**, 4–31 (2011)
19. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **1**, 1–37 (2012)
20. Lorenz, E.N.: Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963)
21. Chen, G., Ueta, T.: Yet another chaotic attractor. *Int. J. Bifurcat. Chaos* **9**, 1465–1466 (1999)
22. Lüand, J., Chen, G.: A new chaotic attractor coined. *Int. J. Bifurcat. Chaos* **12**, 659–661 (2002)