

Parameter estimation of chaotic systems by an oppositional seeker optimization algorithm

Jian Lin · Chang Chen

Received: 14 August 2013 / Accepted: 5 November 2013 / Published online: 12 December 2013
© Springer Science+Business Media Dordrecht 2013

Abstract The parameter estimation can be formulated as a multi-dimensional optimization problem. By combining the seeker optimization algorithm with the opposition-based learning method, an oppositional seeker optimization algorithm is proposed in this work, and is applied to the parameter estimation of chaotic systems. The seeker optimization algorithm provides a new alternative for population-based heuristic search. By considering an estimate and its opposite of current solutions at the same time, the opposition-based learning method is employed for population initialization and also for generation jumping in seeker optimization algorithm. Numerical simulations on two typical chaotic systems are conducted to show the effectiveness and robustness of the proposed scheme.

Keywords Seeker optimization algorithm · Parameter estimation · Chaotic system · Opposition-based learning

J. Lin (✉)
School of Information, Zhejiang University of Finance & Economics, Hangzhou 310018, China
e-mail: linjian@hdu.edu.cn

J. Lin
School of Electrical Engineering, Zhejiang University, Hangzhou 310027, China

C. Chen
School of Media & Design, Hangzhou Dianzi University, Hangzhou 310018, China

1 Introduction

As an interesting complex nonlinear phenomenon, chaos is characterized by a bounded unstable dynamic behavior that exhibits sensitive dependence on the initial state values. In scientific and engineering researches, many nonlinear systems have exhibited phenomenon of chaos. Control and synchronization of chaotic systems have been investigated intensely in a variety of fields during recent years [1–4]. Many methods have been developed to control and synchronize chaotic system, with a condition that parameters of the chaotic system are known in advance. Therefore, the parameter estimation of chaotic systems is important, and many approaches have been proposed for the parameter estimation of chaotic systems [5–10].

Recently, by formulating the parameter estimation of chaotic systems as a multi-dimensional optimization problem, many intelligent optimization schemes have been proposed for such estimation problem. Typically, particle swarm optimization (PSO) in Ref. [11], differential evolution (DE) algorithm in Ref. [12], biogeography-based optimization (BBO) algorithm in Refs. [13, 14]. Among them, DE is with less control parameters and good at exploring global searching, but weak in exploitation and is easy to be trapped in local optima [15]. As a popular evolutionary optimization technique, PSO exhibits significant performance in the initial period of evolution, but might encounter problems in reaching optimum solutions efficiently, and thus the accuracy the algorithm can achieve is limited

[16]. In BBO, poor solutions accept a lot of new features from good ones which helps to improve the quality of those solutions.

Seeker optimization algorithm (SOA) [17] is a newly proposed method concerning searching behaviors of human intelligence. SOA has demonstrated good performance when compared with other evolutionary algorithms (EAs) [18–20]. However, as mentioned by Dai, the author of SOA, this algorithm can be further improved by increasing the diversity of distributed population [21]. In order to enhance the exploration and the exploitation abilities, an oppositional seeker optimization algorithm (OSOA) is devised by combining SOA with opposition-based learning (OBL) [22]. Especially, the diversity of initial population is increased via the uniformly distributed population generating mechanism in OBL. Moreover, new populations during the evolutionary process are also generated with OBL method. By simultaneously considering an estimate and its corresponding opposite estimate, OBL provides a higher chance of finding solutions which are closer to the global optimum. The OSOA is further employed to estimate the parameters of the chaotic system. On the other hand, although the parameter estimations are extensively studied in chaotic systems, few studies [13, 14, 23] have addressed the noise in the systems. Noise will affect the output of the chaotic system and bias the estimation of the original parameters. Based on this consideration, the impact of the additional noise on the parameter-estimation performance of the proposed scheme is analyzed in this study. Simulations on two typical chaotic systems show that the proposed scheme provides a promising candidate for parameter estimation of chaotic systems.

The rest of this paper is organized as follows. In Sect. 2, the problem of parameter estimation for chaotic system is formulated from the view of optimization. In Sect. 3, the OSOA is proposed after briefly introducing SOA and OBL methods. Numerical simulations and conclusions are given in Sects. 4 and 5, respectively.

2 Problem formulation

Considering the following n -dimension chaotic system:

$$\dot{X} = F(X, X_0, \theta), \tag{1}$$

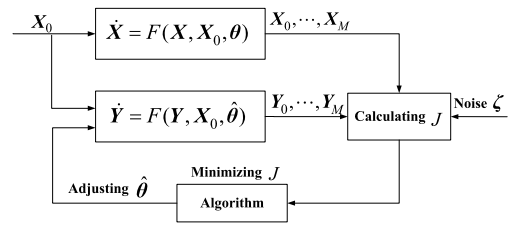


Fig. 1 The principle of parameter estimation for chaotic system

where $X = (x_1, x_2, \dots, x_n)^T \in R^n$ denotes the state vector and X_0 denotes the initial state. $\theta = (\theta_1, \theta_2, \dots, \theta_D)^T$ is a set of original parameters. If the structure of system (1) is known, then the estimated system can be written as

$$\dot{Y} = F(Y, X_0, \hat{\theta}), \tag{2}$$

where $Y = (y_1, y_2, \dots, y_n)^T \in R^n$ is the state vector of the estimated system, $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_D)^T$ is a set of estimated parameters. To cope with the problem of parameter estimation, the following objective function is defined:

$$\min J = \frac{1}{M} \sum_{k=1}^M \|X_k - Y_k + \zeta_k\|^2, \tag{3}$$

where $k = 1, 2, \dots, M$ is the sampling time point and M denotes the length of data used for parameter estimation; ζ_k denotes the additional noise added to the system; X_k and Y_k denote the state vector of the original and the estimated system at time k , respectively.

Clearly, parameter estimation for chaotic system is a multi-dimensional continuous optimization problem, where the decision vector is $\hat{\theta}$, and the optimization goal is to minimize J . The principle of parameter estimation for chaotic system in the view of optimization can be illustrated with Fig. 1.

Obviously, it is difficult to estimate parameters of a chaotic system using traditional optimization methods, one reason being that the chaotic system exhibits dynamic instability and the other that there are multiple local optima in the objective functions. In this paper, an effective oppositional seeker optimization algorithm is designed to find the parameters of system (1). In addition, two indices, as formulated in Eqs. (4) and (5) respectively, are introduced to better

evaluate the performance of the proposed scheme:

$$R_{NSR} = \left[\frac{\sum_{k=1}^M \|\xi_k\|^2}{\sum_{k=1}^M \|X_k\|^2} \right]^{1/2} \times 100 \%, \quad (4)$$

$$F_{EEF} = \left\| \frac{\hat{\theta} - \theta}{\theta} \right\|_2 \times 100 \%, \quad (5)$$

where R_{NSR} is the noise-to-signal value defined as the ratio of noise intensity to the signal intensity in the noise-free system, and F_{EEF} is the error evaluator factor employed to measure the accuracy of the estimated results.

3 Oppositional seeker optimization

An overview of the original seeker optimization algorithm (SOA) is presented in Sect. 3.1, while a brief introduction to the opposition-based learning scheme in Sect. 3.2. The proposed oppositional seeker optimization algorithm is presented in Sect. 3.3.

3.1 Seeker optimization algorithm

SOA [17, 24] is a relatively new population-based heuristic search algorithm. It is based on simulating the act of human search for finding an optimal solution by a seeker population. The whole population is randomly divided into three subpopulations. These subpopulations search over several different domains of the search space. All seekers in the same subpopulation constitute a neighborhood, which represents the social component for social sharing of information.

(1) *Implementation of SOA.* In SOA, for each seeker i , the position update on each variable j is given by the following equation:

$$x_{ij}(t + 1) = x_{ij}(t) + \alpha_{ij}(t) \times d_{ij}(t), \quad (6)$$

where $x_{ij}(t + 1)$ and $x_{ij}(t)$ are the positions of seeker i on the variable j at time steps $t + 1$ and t , respectively; $d_{ij}(t)$ and $\alpha_{ij}(t)$ are search direction and step length of seeker i on the variable j at time step t , where $\alpha_{ij}(t) \geq 0$ and $d_{ij}(t) \in \{-1, 0, 1\}$. Here, i represents the population number and j the variable number to be optimized.

Moreover, seekers in the same subpopulation are searching for the optimal solution using their own information. In order to avoid the convergence of subpopulations trapping into local optima, the position of

the worst seeker of each subpopulation is combined with the best one in each of the other subpopulations using the binomial crossover operator as follows:

$$x_{k_n j, \text{worst}} = \begin{cases} x_{l j, \text{best}} & \text{if } \text{rand}(0, 1) \leq 0.5, \\ x_{k_n j, \text{worst}} & \text{else,} \end{cases} \quad (7)$$

where $\text{rand}(0, 1)$ is a uniformly random real number within $[0, 1]$, $x_{k_n j, \text{worst}}$ is denoted as the j th variable of the n th worst position in the k th subpopulation, $x_{l j, \text{best}}$ is the j th variable of the best position in the l th subpopulation. Here, $n, k, l = 1, 2, \dots, K - 1$ and $k \neq l$. Thus, the diversity of population is increased by sharing good information among subpopulations.

(2) *Search direction.* In SOA, each seeker selects his search direction based on several empirical directions by comparing the current or historical positions of himself or his neighbors. For seeker i , the empirical directions involved are:

$$d_{i, \text{ego}}(t) = \text{sign}(p_{i, \text{best}}(t) - x_i(t)), \quad (8)$$

$$d_{i, \text{alt1}}(t) = \text{sign}(g_{\text{best}}(t) - x_i(t)), \quad (9)$$

$$d_{i, \text{alt2}}(t) = \text{sign}(l_{\text{best}}(t) - x_i(t)), \quad (10)$$

where $d_{i, \text{ego}}(t)$ is egotistic direction, $d_{i, \text{alt1}}(t)$ and $d_{i, \text{alt2}}(t)$ are altruistic directions, $p_{i, \text{best}}(t)$, $g_{\text{best}}(t)$ and $l_{\text{best}}(t)$ represent personal historical best position and neighbors' historical and current best positions, respectively, the function $\text{sign}(\cdot)$ being a signum function on each variable of the input vector. In addition, each seeker i , as an agent, enjoys the properties of proactiveness and exhibits goal-directed behavior [25], which means he may change his search direction in advance according to his past behavior. This behavior is modeled as an empirical direction called proactiveness direction as given in Eq. (11):

$$d_{i, \text{pro}}(t) = \text{sign}(x_i(t_1) - x_i(t_2)), \quad (11)$$

where $t_1, t_2 \in \{t, t - 1, t - 2\}$ and $x_i(t_1)$ is better than $x_i(t_2)$.

Every variable j of $d_i(t)$ is selected applying the following proportional selection rule as in Eq. (12):

$$d_{ij} = \begin{cases} 0, & \text{if } r_j < p_j^{(0)}, \\ +1, & \text{if } p_j^{(0)} \leq r_j \leq p_j^{(0)} + p_j^{(+1)}, \\ -1, & \text{if } p_j^{(0)} + p_j^{(+1)} < r_j \leq 1, \end{cases} \quad (12)$$

where r_j is a uniform random number in $[0, 1]$, $p_j^{(m)}$ ($m \in \{0, +1, -1\}$) is the percent of the numbers of

m from the set $\{d_{ij,ego}, d_{ij,alt1}, d_{ij,alt2}, d_{ij,pro}\}$ on each variable j of all the four empirical directions, i.e., $p_j^{(m)} = (\text{the number of } m)/4$.

(3) *Step length*. In SOA, fuzzy system is adopted to represent the understanding and linguistic behavioral pattern of human searching tendency. The fitness values of all the seekers are sorted in descending manner and turned into the sequence numbers from 1 to S as the inputs of fuzzy reasoning, where S is the size of population. This design renders a fuzzy system to be applicable to a wide range of optimization problems. The calculation expression of step length is presented as

$$\mu_i = \mu_{\max} - \frac{S - I_i}{S - 1}(\mu_{\max} - \mu_{\min}), \tag{13}$$

where I_i is the sequence number of $\mathbf{x}_i(t)$ after sorting the fitness values, μ_{\max} is the maximum membership degree value which is equal to or a little less than 1. Bell membership function $\mu(x) = e^{-x^2/2\delta^2}$ is well utilized in the action part of fuzzy reasoning, and the parameter δ in the function is determined as follows:

$$\delta = \omega \times \text{abs}(\mathbf{x}_{\text{best}} - \mathbf{x}_{\text{rand}}), \tag{14}$$

where the symbol $\text{abs}(\cdot)$ produces an absolute value in response to input vector, ω is used to decrease the step length with time step increasing so as to gradually improve the search precision. The \mathbf{x}_{best} and \mathbf{x}_{rand} are the best seeker and a randomly selected seeker, respectively, from the same subpopulation to which the i th seeker belongs, and \mathbf{x}_{rand} is different from \mathbf{x}_{best} . The step length $\alpha_{ij}(t)$ for every variable j is given as

$$\alpha_{ij}(t) = \delta_j \sqrt{-\ln(\text{rand}(\mu_i, 1))}, \tag{15}$$

where $\text{rand}(\mu_i, 1)$ returns a uniformly random real number within $[\mu_i, 1]$.

3.2 Opposition-based learning

Opposition-based learning (OBL) has been used by EAs to improve the convergence [22, 26–28]. The main idea of OBL is to consider an estimate as well as its corresponding opposite estimate to achieve a better approximation of the current candidate solutions.

We assume that $\mathbf{X} = (x_1, x_2, \dots, x_n)$ is a point in an n -dimensional space, where $x_j \in R$ and $x_j \in [a_j, b_j] \forall j \in \{1, 2, \dots, n\}$; then the opposite point

$\mathbf{X}' = (x'_1, x'_2, \dots, x'_n)$ is denoted by its elements as $x'_j = a_j + b_j - x_j$.

Now, by employing the opposite point denotation, the OBL can be used in optimization as:

- Generate a point $\mathbf{X} = (x_1, x_2, \dots, x_n)$ and its opposite point $\mathbf{X}' = (x'_1, x'_2, \dots, x'_n)$ in an n -dimensional search space.
- Evaluate the fitness of both points $f(\mathbf{X})$ and $f(\mathbf{X}')$.
- If $f(\mathbf{X}') \leq f(\mathbf{X})$ (for minimization problem, vice versa), then replace \mathbf{X} with \mathbf{X}' ; otherwise, continue with \mathbf{X} .

Thus, we see that the point and its opposite point are evaluated simultaneously in order to obtain the fitter one.

3.3 OSOA approach

As an extension of the original SOA approach, an oppositional seeker optimization algorithm (OSOA) is proposed in this paper. The OBL idea is embedded in SOA to improve its performance. On the one hand, the OBL approach is utilized to increase the diversity of initial population. On the other hand, new populations are also generated during the evolutionary process to accelerate its convergence speed. By fusing the operators in SOA with OBL technique, the exploration and exploitation capability may be enhanced and well balanced, and thus result in a more effective algorithm for global optimization. The main procedure of the OSOA is described in Table 1 and can be described as follows.

(1) *Population initialization*. In OSOA, initial population is generated via OBL. A population of $2S$ positions is initialized by using uniform random distribution and their quasi-opposite solutions, which is shown as

$$x_{ij} \leftarrow a_j + (b_j - a_j) \times \text{rand}(0, 1), \tag{16}$$

$$x_{(S+i)j} \leftarrow a_j + b_j - x_{ij}, \tag{17}$$

where a_j and b_j are the lower and upper bounds of j th variable, respectively; x_{ij} and $x_{(S+i)j}$ are the positions of seeker i and $S + i$ on the j th variable, out of which S best positions are selected.

(2) *Generation jumping*. Unlike OBL-based initialization, generation jumping during the evolutionary process calculates the opposite population dynamically. Instead of using variables' predefined interval boundaries $[a_i, b_i]$, generation jumping calculates the

Table 1 The main procedure of OSOA

The time step $t \leftarrow 0$;
 Generating S positions in the search space using OBL method by Eqs. (16) and (17);
 Computing the objective function value for these S positions;
 Saving the historical best fitness value and its position;
 Randomly divided the whole population into three subpopulations;
While the termination condition is not satisfied **do**
 Updating the time step $t \leftarrow t + 1$;
 For each seeker
 Computing the direction $d_i(t)$ by Eqs. (8)–(12);
 Computing the step length $\alpha_i(t)$ by Eqs. (13)–(15);
 Updating each seeker’s position by Eq (6);
 Re-computing the fitness value for each seeker;
 Next for
 Updating the historical best position among the populations;
 Generation jumping in the dynamical search space using OBL method by Eq. (18);
End while

opposite of each variable based on minimum and maximum values of that variable in the current population:

$$x_{(S+i)j} \leftarrow x_{\min}^P(j) + x_{\max}^P(j) - x_{ij}, \tag{18}$$

where $x_{\min}^P(j)$ and $x_{\max}^P(j)$ are the minimum and maximum values of the j th variable in current population P , respectively.

4 Simulations and comparisons

To verify the effectiveness of the proposed scheme, two typical chaotic systems are employed as testbeds in this section. The simulation was done using MATLAB 7.1 on Core 2 Duo processor, 2.26 GHz with 2 GB RAM. The parameters used in SOAs are set as follows: population size S is 40, the maximum value μ_{\max} and minimum value μ_{\min} of membership degree are set to 0.99 and 0.0111, respectively.

4.1 On the Lorenz system

As a typical chaotic system, Lorenz system is employed as the first example, which can be described as:

$$\begin{cases} \dot{x} = \delta(y - x), \\ \dot{y} = \gamma x - xz - y, \\ \dot{z} = xy - bz, \end{cases} \tag{19}$$

where $\delta = 10$, $\gamma = 28$ and $b = 8/3$ is the original setting for the system parameters.

The system is initialized as state X_0 , which is randomly selected from its evolution process. The length of the sampled data M is set as 300. Moreover, the proposed scheme is compared with original SOA, PSO [11] and HBBO [13]. For a fair comparison, the maximum number of generations is set to 100, as used for PSO and HBBO in the literature [11, 13]. Table 2 lists the statistical results of the best, the mean and the worst estimated parameters for Lorenz system over 20 independent runs. It can be seen that the best fitness values obtained by OSOA are better than those by SOA, PSO and HBBO, and the same is for the mean and worst values in the table. Even the worst results by OSOA still outperform the best ones of the other three algorithms. The averaged result of the evolving process of the estimated parameters and the fitness values by SOA and OSOA are shown in Fig. 2. From the figure it can be seen that OSOA converges to the optimal solution rapidly, and has faster convergence speed than the SOA. Therefore, OSOA demonstrates better effectiveness and robustness than SOA, PSO and HBBO for parameter estimation of the well-known Lorenz system.

To better simulate the real world, white Gaussian noise is added to each state variable, and the running parameters are kept unchanged. The averaged results by OSOA over 20 independent runs are given in Table 3. It is obvious that OSOA still can achieve good

Table 2 Statistical results of different methods for Lorenz system

Algorithm		OSOA	SOA	PSO	HBBO
Best	δ	10.0000	10.0000	9.99533	10.0000
	γ	28.0000	28.0000	28.0071	28.0000
	b	2.66667	2.66667	2.66701	2.66667
	J	0	3.91e-22	4.86e-2	2.08e-10
Mean	δ	10.0000	10.0000	10.0184	10.0013
	γ	28.0000	28.0000	27.9939	28.0000
	b	2.66667	2.66667	2.66628	2.66669
	J	7.69e-30	1.31e-19	4.18e+0	9.48e-5
Worst	δ	10.0000	10.0000	10.6082	9.92285
	γ	28.0000	28.0000	27.7044	28.0141
	b	2.66667	2.66667	2.65723	2.66473
	J	4.09e-29	1.03e-18	3.94e+1	9.71e-4

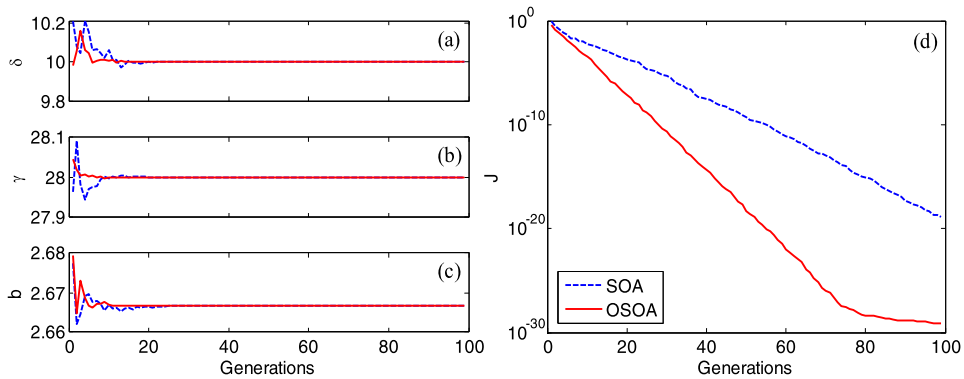


Fig. 2 Evolving process of parameter estimation for Lorenz system by SOA and OSOA: (a) searching process for δ , (b) searching process for γ , (c) searching process for b , (d) convergence trajectories of fitness value

Table 3 Statistical results obtained by OSOA for Lorenz system with different noise

R_{NSR}	1 %	10 %	25 %	50 %
δ	9.90991	9.94954	10.4190	10.0306
γ	2.66520	2.63934	2.69398	2.86334
b	28.0113	28.0451	27.7936	24.7042
F_{EEF}	0.90 %	1.15 %	4.38 %	13.89 %

results when the noise is concerned; in particular, if the value of R_{NSR} is small enough, say, 1 %, an ideal outcome may be obtained with $F_{EEF} = 0.9\%$. In addition, if R_{NSR} is substantially less than 25 %, an overall satisfactory outcome, with $F_{EEF} < 5\%$, can be obtained. However, the obtained result is relatively poor, with $F_{EEF} = 13.89\%$, under the noise of $R_{NSR} = 50\%$.

4.2 On the Chen systems

To further evaluate the proposed scheme, the Chen’s chaotic system is adopted, which is described as

$$\begin{cases} \dot{x} = \delta(y - x), \\ \dot{y} = (b - \delta)x - xz + by, \\ \dot{z} = xy - \gamma z, \end{cases} \quad (20)$$

Table 4 Statistical results of different methods for Chen system

Algorithm		OSOA	SOA	DEA
Best	δ	35.0000	35.0000	34.9312
	γ	3.00000	3.00000	3.06700
	b	28.0000	28.0000	27.9338
	J	1.29e-21	1.06e-11	6.03e-3
Mean	δ	35.0000	35.0000	36.1181
	γ	3.00000	3.00004	3.05838
	b	28.0000	28.0000	28.5292
	J	1.53e-20	8.55e-11	8.98e-1
Worst	δ	35.0000	35.0000	37.9664
	γ	3.00000	3.00016	3.12150
	b	28.0000	28.0000	29.3849
	J	4.32e-20	3.53e-10	3.27e+0

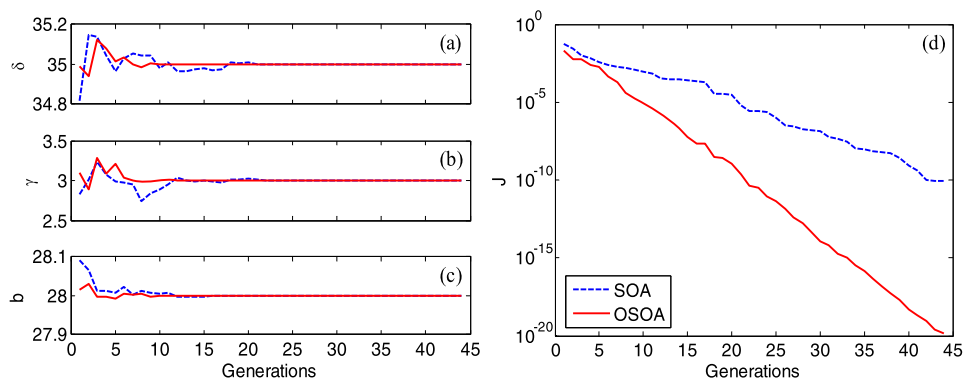


Fig. 3 Evolving process of parameter estimation for the Chen system by SOA and OSOA: (a) searching process for δ , (b) searching process for γ , (c) searching process for b , (d) convergence trajectories of fitness value

with $\delta = 35$, $\gamma = 3$ and $b = 28$ as the original parameter setting.

The statistical results of the estimated parameters of the Chen system are summarized in Table 4, where the results of DEA given in Ref. [29] are used for comparison. To make a fair comparison, the maximum generation number is set to 45. In addition, totally 5 runs are made for each algorithm and the length of sampled data M is set to 100. From Table 4 it can be seen that OSOA have more accurate results than SOA and DEA. Figure 3 depicts the convergence profile of the estimated parameters and fitness values of OSOA and SOA; obviously, OSOA converges to the optimal solution more rapidly than SOA. Table 5 presents the results under the influence of noise with different R_{NSR}

values. It can be observed that the results are satisfactory in the cases with R_{NSR} value smaller than 10 %, especially a fairly good outcome can be obtained if the R_{NSR} value reaches 1 %.

It can be concluded that OSOA is significantly better and statistically more robust than all the other listed algorithms in terms of search performance.

5 Conclusions

In this paper, parameter estimation of chaotic system is formulated as a multi-dimensional optimization problem from the viewpoint of optimization. An OSOA scheme is proposed based on OBL method and applied to solve this optimization problem. Numerical

Table 5 Statistical results of OSOA for the Chen system with different noise

R_{NSR}	1 %	10 %	25 %	50 %
δ	34.9904	35.0705	33.5226	35.8658
γ	2.96413	3.25937	2.42803	3.92678
b	28.0087	28.0248	28.1566	28.2010
F_{EEF}	1.20 %	8.65 %	19.54 %	31.00 %

simulations based on Lorenz and Chen chaotic systems and comparisons with some typical existing approaches demonstrated the effectiveness and robustness of the proposed scheme. The future research is to apply this scheme to some other systems, such as high-dimensional, dynamical and uncertain chaotic systems.

Acknowledgement This work is supported in part by the Zhejiang Provincial Educational Committee of China under Grant No. Y201016676. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

- Shieh, C.S., Hung, R.T.: Hybrid control for synchronizing a chaotic system. *Appl. Math. Model.* **35**(8), 3751–3758 (2011)
- Pan, L., Zhou, W., Zhou, L., Sun, K.: Chaos synchronization between two different fractional-order hyperchaotic systems. *Communications in Nonlinear Science and Numerical Simulation* **16**(6), 2628–2640 (2011)
- Santos Coelho, L., de Andrade Bernert, D.L.: An improved harmony search algorithm for synchronization of discrete-time chaotic systems. *Chaos Solitons Fractals* **41**(5), 2526–2532 (2009)
- Pisarchik, A.N., Ruiz-Oliveras, F.R.: Optical chaotic communication using generalized and complete synchronization. *IEEE J. Quantum Electron.* **46**(3), 279–284 (2010)
- Chang, J.F., Yang, Y.S., Liao, T.L., Yan, J.J.: Parameter identification of chaotic systems using evolutionary programming approach. *Expert Syst. Appl.* **35**(4), 2074–2079 (2008)
- Fotsin, H.B., Woafu, P.: Adaptive synchronization of a modified and uncertain chaotic Van der Pol-Duffing oscillator based on parameter identification. *Chaos Solitons Fractals* **24**(5), 1363–1371 (2005)
- Maybhate, A., Amritkar, R.: Use of synchronization and adaptive control in parameter estimation from a time series. *Phys. Rev. E, Stat. Nonlinear Soft Matter Phys.* **59**(1), 284 (1999)
- Saha, P., Banerjee, S., Roy Chowdhury, A.: Chaos, signal communication and parameter estimation. *Phys. Lett. A* **326**(1), 133–139 (2004)
- Wu, X., Hu, H., Zhang, B.: Parameter estimation only from the symbolic sequences generated by chaos system. *Chaos Solitons Fractals* **22**(2), 359–366 (2004)
- Xu, D.L., Lu, F.F.: An approach of parameter estimation for non-synchronous systems. *Chaos Solitons Fractals* **25**(2), 361–366 (2005)
- He, Q., Wang, L., Liu, B.: Parameter estimation for chaotic systems by particle swarm optimization. *Chaos Solitons Fractals* **34**(2), 654–661 (2007)
- Peng, B., Liu, B., Zhang, F.Y., Wang, L.: Differential evolution algorithm-based parameter estimation for chaotic systems. *Chaos Solitons Fractals* **39**(5), 2110–2118 (2009)
- Lin, J., Xu, L.: Parameter estimation for chaotic systems based on hybrid biogeography-based optimization. *Acta Phys. Sin.* **62**(3), 030505 (2013)
- Wang, L., Xu, Y.: An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems. *Expert Syst. Appl.* **38**(12), 15103–15109 (2012)
- Rao, R., Savsani, V., Balic, J.: Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. *Eng. Optim.* **44**(12), 1447–1462 (2012)
- Yang, X.M., Yuan, J.S., Yuan, J.Y., Mao, H.N.: A modified particle swarm optimizer with dynamic adaptation. *Appl. Comput. Math.* **189**(2), 1205–1213 (2007)
- Dai, C., Chen, W., Song, Y., Zhu, Y.: Seeker optimization algorithm: a novel stochastic search algorithm for global numerical optimization. *J. Syst. Eng. Electron.* **21**(2), 300–311 (2010)
- Banerjee, A., Mukherjee, V., Ghoshal, S.: Seeker optimization algorithm for load-tracking performance of an autonomous power system. *Int. J. Electr. Power Energy Syst.* **43**(1), 1162–1170 (2012)
- Dai, C.H., Chen, W.R., Zhu, Y.F., Jiang, Z.L., You, Z.Y.: Seeker optimization algorithm for tuning the structure and parameters of neural networks. *Neurocomputing* **74**(6), 876–883 (2011)
- Shaw, B., Mukherjee, V., Ghoshal, S.: Solution of economic dispatch problems by seeker optimization algorithm. *Expert Syst. Appl.* **39**(1), 508–519 (2012)
- Dai, C.: Seeker optimization algorithm and its applications. Ph.D. dissertation, Southwest Jiaotong University (2009)
- Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: *International Conference on Computational Intelligence for Modelling Control and Automation*, Vienna, Austria, 28–30 November, pp. 695–701. IEEE Press, Los Alamitos (2005)
- Li, N.Q., Pan, W., Yan, L.S., Luo, B., Xu, M.F., Jiang, N.: Parameter estimation for chaotic systems with and without noise using differential evolution-based method. *Chin. Phys. B* **20**(6), 060502 (2011)
- Dai, C., Zhu, Y., Chen, W.: Seeker optimization algorithm. In: *Computational Intelligence and Security*, pp. 167–176 (2007)

25. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. *Knowl. Eng. Rev.* **10**(2), 115–152 (1995)
26. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* **12**(1), 64–79 (2008)
27. Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., Ventresca, M.: Enhancing particle swarm optimization using generalized opposition-based learning. *Inf. Sci.* **181**(20), 4699–4714 (2011)
28. Wang, J., Wu, Z., Wang, H.: Hybrid differential evolution algorithm with chaos and generalized opposition-based learning. *Adv. Comput. Intell.* **6382**, 103–111 (2010)
29. Ho, W.H., Chou, J.H., Guo, C.Y.: Parameter identification of chaotic systems using improved differential evolution algorithm. *Nonlinear Dyn.* **61**(1), 29–41 (2010)