Soft Comput (2018) 22:8317–8339
https://doi.org/10.1007/s00500-017-2779-0

METHODOLOGIES AND APPLICATION

Parameter identification of chaotic systems using a shuffled backtracking search optimization algorithm

Morteza Alinia Ahandani[1] · Amir Rikhtehgar Ghiasi[1] · Hamed Kharrati[1]

Published online: 23 August 2017

Abstract An accurate mathematical model has a vital role in controlling and synchronization of chaotic systems. But generally in real-world problems, parameters are mixed with mismatches and distortions. This paper proposes two simple but effective estimation methods to detect the unknown parameters of chaotic models. These methods focus on improving the performance of a recently proposed evolutionary algorithm called backtracking search optimization algorithm (BSA). In this research firstly, a new operator to generate initial trial population is proposed. Then a group search ability is provided for the BSA by proposing a shuffled BSA (SBSA). Grouping population into several sets can provide a better exploration of search space, and an independent local search of each group increases exploitation ability of the BSA. Also new proposed operator to generate initial trial population, by providing a deep search, increases considerably the quality of solutions. The superiority of the proposed algorithms is investigated on parameter identification of 10 typical chaotic systems. Practical experiences and nonparametric analysis of obtained results show that both of the proposed ideas to improve performance of original BSA are very effective and robust so that the BSA by aforementioned ideas produces similar and promising results over repeated runs. A considerably better performance of proposed algorithms based on average of objective functions demonstrates that the proposed ideas can evolve robustness and consistence of BSA. A comparison of the proposed algorithms in this study with respect to other algorithms reported in the literature confirms a considerably better performance of proposed algorithms.

Keywords Chaotic models · Backtracking search optimization algorithm · Group search · Shuffled BSA

Communicated by V. Loia.

✉ Morteza Alinia Ahandani
alinia@tabrizu.ac.ir

Amir Rikhtehgar Ghiasi
agiasi@tabrizu.ac.ir

Hamed Kharrati
kharrati@tabrizu.ac.ir

1 Department of Control Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

1 Introduction

Chaos is a common feature in nonlinear dynamical systems. Nonlinear systems which have chaos feature are highly sensitive to initial conditions so that a small change in initial conditions of dynamic system yields widely diverging outcomes. It causes an infinite number of unstable periodic motions, so behavior of nonlinear systems becomes unpredictable and complex to be analyzed. Chaotic behavior exists in many real-world systems and phenomena such as biological systems, for example chaotic behavior in the population growth or in epileptic brain seizures, ecological systems, for example chaotic model for hydrology, economic and financial systems, for example improving economic models, transportation and traffic systems, for example traffic forecasting, chemical reactions, for example peroxidase–oxidase reaction and electrical engineering, for example chaotic oscillators. However, all the aforementioned systems and phenomena are stochastic and even unpredictable, but they are actually deterministic in nature and they can be predicted and controllable by natural laws if mathematical models of them are successfully constructed.

The traditional trend of analyzing and understanding chaos has already evolved a new phase in investigation:

controlling and utilizing chaos where detecting the unstable periodic orbits and estimating the unknown parameters of the chaos are of vital importance (Gao et al. 2013). So for controlling and synchronization of chaotic systems, a mathematical model is vital. Since such models have been applied in definite chaotic systems with predetermined system parameters, however, there generally exist parameter mismatches and distortions in real-world problems. Therefore, this topic has become popular among the researchers, and numerous scientific studies have been proposed to overcome this drawback by suggesting novel solution (Wang and Xu 2011).

Parameter estimation of chaotic systems can be modeled as a multidimensional optimization problem after defining an appropriate fitness function. Since in this problem, the structure of system model is known in advance, the optimization methods should optimize a fitness function which is a difference between output of the true system and the estimated model under the same inputs. By this modeling, a variety of optimization methods can be applied on it to extract its unknown parameters. This issue has been becoming topic of many researches during the past two decades (Konnur 2003; Park and Kwon 2005; Zaher 2008; Yu et al. 2009; Sun et al. 2009; Li et al. 2011). The original parameters of a chaotic system are not easy to estimate because of the unstable dynamic of the chaotic system. Meanwhile, it is very difficult for traditional mathematical methods to identify the true values of those parameters to achieve global optimization, since there are lots of local optima in the landscape of the goal function. Nowadays, the development of effective approaches for solving parameter estimation is still a hot topic with significance in both academic and engineering fields (Wang and Xu 2011).

Tendency of recent researches for parameter estimation of chaotic systems has been propelled to heuristic algorithms especially with stochastic search techniques such as evolutionary algorithms (EAs). The EAs have a prominent advantage over other types of numerical methods. They only require information about the objective function itself, which can be either explicit or implicit. Other accessory properties such as differentiability or continuity are not necessary. As such, they are more flexible in dealing with a wide spectrum of problems (Brest et al. 2006).

A wide variety of EAs are progressively being applied on parameter estimation of chaotic systems in recent years such as differential evolution (DE) (Tang et al. 2012; Gao et al. 2014), particle swarm optimization (PSO) (Yuan and Yang 2012; Chen et al. 2014), biogeography-based optimization (BBO) (Wang and Xu 2011; Lin 2014) and artificial bee colony (Gao et al. 2012; Hu et al. 2015).

One of the very recently proposed population-based EAs is the backtracking search optimization algorithm (BSA) (Civicioglu 2013). Civicioglu (2013) in an attempt to develop a simpler and more effective search algorithm and to mitigate the effects of problems that are frequently encountered in EAs, such as excessive sensitivity to control parameters, premature convergence, and slow computation, proposed the BSA. It has only a single control parameter which the BSA is not oversensitive to the initial value of this parameter. By employing a memory, this algorithm allows to take advantage of experiences gained from previous generations when it generates a trial preparation. After Civicioglu (2013), only one another improved version of this algorithm was proposed in different literature. Lin (2015) proposed an opposition-based version of BSA for parameter identification of hyperchaotic systems. This new version of BSA is trying to increase the diversity of initial population and to accelerate the convergence speed. The current research is an attempt to provide a grouping and parallel search ability for the BSA algorithm. This idea was borrowed from shuffled frog leaping algorithm (SFLA) (Eusuff and Lansey 2003). Based on this idea, a population is divided into several groups and then each of these groups tries to evolve itself using a BSA evolutionary process. After a predefined repetition of this evolutionary process, all groups are shuffled together and new groups are made and this process is repeated again. Grouping population into several sets can provide a better exploration of search space, and an independent local search increases exploitation ability of BSA. Also a possibility of rebuilding groups with new members causes participating each of members in previous experience of all other members. Another concept to evolve exploitation ability of BSA in this research is to propose a new operator to generate initial trial population. This new operator provides a deep search around found promising areas.

The rest of the paper is organized as follows. In the next section, chaotic systems and dynamic equations of employed typical systems are described. In Sect. 3, the BSA is briefly presented. In Sect. 4, the utilized strategy to improve the BSA is discussed. The simulation results are presented and analyzed in Sect. 5. Section 6 concludes the paper.

2 Chaotic systems

In this section firstly, the chaotic systems are briefly described. Then problem of parameter identification of chaotic systems is formulated as an optimization problem. In the last part of this section, dynamic equations of 10 typical chaotic systems are described.

2.1 Chaotic system description and problem formulation

Generally, chaotic systems are nonlinear deterministic systems. Those display complex, noisy-like and unpredictable behaviors. The sensitive dependency on both initial conditions and parameter variations is a prominent characteristic of chaotic behavior. A general form for chaotic systems is given as follows (Jiang et al. 2015):

$$\dot{x}(t) = f(x(t), x(t-\tau), x_0, \theta) \tag{1}$$

where $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^{\mathrm{T}} \in R^n$ denotes the state vectors of chaotic system, $\dot{x}(t)$ denotes the derivative of the state vector $x(t)$, $x(t-\tau)$ denotes delay vector (τ is the delay), $x_0 = [x_{01}, x_{02}, \ldots, x_{0n}]^{\mathrm{T}}$ is the initial vector, and $\theta = [\theta_1, \theta_2, \ldots, \theta_m]^{\mathrm{T}} \in R^m$ are unknown parameters, and suppose the form of nonlinear vector function $f : R^n \times R^m \to R^n$ is known and f is continuously differentiable.

Since the structure of system model is known in advance, the estimated system can be depicted as follows:

$$\dot{\hat{x}}(t) = f(\hat{x}(t), \hat{x}(t-\hat{\tau}), x_0, \hat{\theta}) \tag{2}$$

where $\hat{x}(t) = [\hat{x}_1(t), \hat{x}_2(t), \ldots, \hat{x}_n(t)]^{\mathrm{T}} \in R^n$ denotes the state vectors, $\hat{\tau}$ and $\hat{\theta} = \left[\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_m\right]^{\mathrm{T}} \in R^m$ are the identification of unknown parameter τ and θ, respectively.

The basic principle of parameter estimation is to compare the output of the true system and the estimated model under the same inputs and to adjust the parameters $\theta = [\theta_1, \theta_2, \ldots, \theta_m]^{\mathrm{T}} \in R^m$ for minimizing a predefined error function for a number of given samples, e.g., the following sum square error (SSE) function.

$$\mathrm{SSE} = \sum_{k=1}^{L} \left\| x(k) - \hat{x}(k) \right\|^2 \tag{3}$$

where $\hat{x}(k)$ is the output of the model with estimated parameters, L denotes the total number of sampling points, and $\|.\|$ represents the Euclidean norm of vectors. So in an overall view, the problem of parameters identification for a chaotic system to be solved using an optimization method can be formulated as follows:

$$\begin{aligned} &\min J = \sum_{k=1}^{L} \left\| x(k) - \hat{x}(k) \right\|^2 \\ &\text{subject to} \begin{cases} \dot{x}(t) = f(x(t), x(t-\tau), x_0, \theta) \\ \dot{\hat{x}}(t) = f(\hat{x}(t), \hat{x}(t-\hat{\tau}), x_0, \hat{\theta}) \\ L_i \le \hat{\theta}_i \le U_i i = 1, \ldots, m \\ \hat{\tau}_{\min} \le \hat{\tau} \le \hat{\tau}_{\max} \end{cases} \end{aligned} \tag{4}$$

$(\hat{\theta}, \hat{\tau})$ is decision vector, and the optimization goal is to minimize J. L_i and U_i correspond to the upper and lower boundary of $\hat{\theta}_i$, respectively. Also $\hat{\tau}_{\min}$ and $\hat{\tau}_{\max}$ are the upper and lower boundary of $\hat{\tau}$, respectively (Jiang et al. 2015).

The parameter estimation of chaotic systems is not easy because of the unstable dynamic of the chaotic systems. Moreover, due to multiple variables in the problem and multiple local search optima in the landscape of the objective functions, traditional optimization can easily trap in local optima. Nowadays, the development of effective approaches for solving parameter identification is still an active research subject with significance in both academic and engineering fields (Wang and Xu 2011). So this research proposes two new versions of BSA for efficiently solving this problem.

2.2 Dynamic equations of typical chaotic systems

This section describes dynamic equations of 10 typical chaotic systems. To the best our knowledge, most of the papers applied on parameter identification of chaotic systems employed at most three systems to investigate their proposed methods. This research to provide a better investigation about performance of proposed algorithms uses 10 typical chaotic systems. Also since characteristics of all systems such as number of unknown parameters, dynamic range of parameters, initial conditions and number of samples are definite, so theses sets of typical chaotic systems can be used as a criterion to investigate and compare performance of different algorithms in the future researches.

Example 1 Lorenz chaotic system.

$$\begin{aligned} &\dot{x}_1(t) = \theta_1(x_2(t) - x_1(t)) \\ &\dot{x}_2(t) = \theta_2 x_1(t) - x_2(t) - x_1(t)x_3(t) \\ &\dot{x}_3(t) = x_1(t)x_2(t) + \theta_3 x_3(t) \\ &x_1(0) = 0.1, x_2(0) = 0.1x_3(0) = 0.1 \\ &\quad t = 1, 2, \ldots, 100 \end{aligned} \tag{5}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3] = [10, 28, 2.6667]$. In addition, the searching ranges are set as follows: $5 \le \theta_1 \le 15$, $20 \le \theta_2 \le 30$ and $0.1 \le \theta_3 \le 10$.

Example 2 Chen chaotic system.

$$\begin{aligned} &\dot{x}_1(t) = \theta_1(x_2(t) - x_1(t)) \\ &\dot{x}_2(t) = \theta_4 x_1(t) - x_1(t)x_3(t) + \theta_3 x_2(t) \\ &\dot{x}_3(t) = x_1(t)x_2(t) - \theta_2 x_3(t) \\ &x_1(0) = -9, x_2(0) = -5, x_3(0) = 14 \\ &\quad t = 1, 2, \ldots, 100 \end{aligned} \tag{6}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3, \theta_4]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3, \theta_4] = [35, 3, 28, -7]$. In addition, the searching ranges are set as follows: $30 \le \theta_1 \le 40$, $0.1 \le \theta_2 \le 10$, $20 \le \theta_3 \le 30$ and $-10 \le \theta_4 \le -0.1$.

Example 3 Rossler chaotic system.

$$\begin{aligned}
\dot{x}_1(t) &= -x_2(t) - x_3(t)\\
\dot{x}_2(t) &= x_1(t) + \theta_1 x_2(t)\\
\dot{x}_3(t) &= \theta_2 + x_1(t)x_3(t) - \theta_3 x_3(t)\\
x_1(0) &= 0.5, x_2(0) = 1.5, x_3(0) = 0.1\\
t &= 1, 2, \ldots, 100 \qquad (7)
\end{aligned}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3] = [0.5, 0.2, 10]$. In addition, the searching ranges are set as follows: $0.1 \le \theta_1 \le 1$, $0.1 \le \theta_2 \le 1$ and $5 \le \theta_3 \le 15$.

Example 4 Arneodo chaotic system.

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t)\\
\dot{x}_2(t) &= x_3(t)\\
\dot{x}_3(t) &= -\theta_1 x_1(t) - \theta_2 x_2(t) - \theta_3 x_3(t) + \theta_4 x_1^3(t)\\
x_1(0) &= -0.2, x_2(0) = 0.5, x_3(0) = 0.2\\
t &= 1, 2, \ldots, 100 \qquad (8)
\end{aligned}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3, \theta_4]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3, \theta_4] = [-5.5, 3.5, 0.8, -1.0]$. In addition, the searching ranges are set as follows: $-6 \le \theta_1 \le -5$, $2 \le \theta_2 \le 5$, $0.1 \le \theta_3 \le 1$ and $-1.5 \le \theta_4 \le -0.5$.

Example 5 Duffing chaotic system.

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t)\\
\dot{x}_2(t) &= x_1(t) - x_1^3(t) - \theta_1 x_2(t) + \theta_2 \cos(\theta_3 t)\\
x_1(0) &= 0.21, x_2(0) = 0.31\\
t &= 1, 2, \ldots, 100 \qquad (9)
\end{aligned}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3] = [0.15, 0.31, 1]$. In addition, the searching ranges are set as follows: $0.1 \le \theta_1 \le 1$, $0.1 \le \theta_2 \le 1$ and $0.1 \le \theta_3 \le 2$.

Example 6 Genesio-Tesi chaotic system.

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t)\\
\dot{x}_2(t) &= x_3(t)\\
\dot{x}_3(t) &= -\theta_1 x_1(t) - \theta_2 x_2(t) - \theta_3 x_3(t) + \theta_4 x_1^2(t)\\
x_1(0) &= -0.1, x_2(0) = 0.5, x_3(0) = 0.2\\
t &= 1, 2, \ldots, 100 \qquad (10)
\end{aligned}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3, \theta_4]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3, \theta_4] = [1.1, 1.1, 0.45, 1.0]$. In addition, the searching ranges are set as follows: $1 \le \theta_1 \le 2$, $1 \le \theta_2 \le 2$, $0.1 \le \theta_3 \le 1$ and $0.1 \le \theta_4 \le 1.5$.

Example 7 Financial chaotic system.

$$\begin{aligned}
\dot{x}_1(t) &= x_3(t) + x_1(t)(x_2(t) - \theta_1)\\
\dot{x}_2(t) &= 1 - \theta_2 x_2(t) - x_1^2(t)\\
\dot{x}_3(t) &= -x_1(t) - \theta_3 x_3(t)\\
x_1(0) &= 2, x_2(0) = -1, x_3(0) = 1\\
t &= 1, 2, \ldots, 100 \qquad (11)
\end{aligned}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3] = [1, 0.1, 1]$. In addition, the searching ranges are set as follows: $0.5 \le \theta_1 \le 1.5$, $0.01 \le \theta_2 \le 1$ and $0.5 \le \theta_3 \le 1.5$.

Example 8 Lu chaotic system.

$$\begin{aligned}
\dot{x}_1(t) &= \theta_1(x_2(t) - x_1(t))\\
\dot{x}_2(t) &= x_2(t) - x_1^2(t)\\
\dot{x}_3(t) &= x_1(t)x_2(t) - \theta_2 x_3(t)\\
x_1(0) &= 0.2, x_2(0) = 0.5, x_3(0) = 0.3\\
t &= 1, 2, \ldots, 100 \qquad (12)
\end{aligned}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3] = [36, 3, 20]$. In addition, the searching ranges are set as follows: $30 \le \theta_1 \le 40$, $0.1 \le \theta_2 \le 10$ and $15 \le \theta_3 \le 25$.

Example 9 Chuas oscillator chaotic system.

$$\begin{aligned}
\dot{x}_1(t) &= \theta_1(x_2(t) - x_1(t) + \theta_4 x_1(t) - W(x_4(t))x_1(t))\\
\dot{x}_2(t) &= x_1(t) - x_2(t) + x_3(t)\\
\dot{x}_3(t) &= -\theta_2 x_2(t) - \theta_3 x_3(t)\\
\dot{x}_4(t) &= x_1(t)\\
W(x_1(t)) &= \begin{cases} \theta_5 : |x_1(t)| < 1\\ \theta_6 : |x_1(t)| > 1 \end{cases}\\
x_1(0) &= 0.8, x_2(0) = 0.05, x_3(0) = 0.007, x_4(0) = 0.6\\
t &= 1, 2, \ldots, 100 \qquad (13)
\end{aligned}$$

where the parameters to be estimated are $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [10.725, 10.593, 0.268, -0.7872, -1.1726]$. In addition, the searching ranges are set as follows: $5 \le \theta_1 \le 10$, $10 \le \theta_2 \le 20$, $0.1 \le \theta_3 \le 1$, $0.1 \le \theta_4 \le 2$, $0.3 \le \theta_5 \le 3$ and $0.1 \le \theta_6 \le 1$.

Example 10 Henon chaotic system.

$$\begin{aligned}
&\dot{x}_1(t) = x_2(t) + 1 - \theta_1 x_1^2(t)\\
&\dot{x}_2(t) = \theta_2 x_1(t)\\
&x_1(0) = 0.8, x_2(0) = 0.5\\
&\quad t = 1, 2, \ldots, 100
\end{aligned} \tag{14}$$

where the parameters to be estimated are $[\theta_1, \theta_2]$. In this example, the real system parameters are assumed to be $[\theta_1, \theta_2] = [1.4, 0.3]$. In addition, the searching ranges are set as follows: $1 \leq \theta_1 \leq 3$ and $0 \leq \theta_2 \leq 2$.

Figure 1 shows the phase portraits of aforementioned systems to show their chaotic behavior.

3 Backtracking search optimization algorithm

Backtracking search optimization algorithm (BSA) is a new population-based EA which uses well-known operators of genetic algorithms (GAs) in a new structure. In addition to these operators, i.e., selection, mutation and crossover, several unique mechanisms are inserted in BSA such as a memory in which it stores a population from a randomly chosen previous generation. Based on Civicioglu (2013), the BSA has five main steps including: initialization, selection-I, mutation, crossover and selection-II. These five steps are formulated as follows:

(1) Initialization
The BSA starts with a random initial sampling of individuals within the search space using a uniform distribution:

$$\begin{aligned}
&POP_{i,j} = U(low_j, up_j)\\
&\quad \text{for } i = 1, \ldots, N_{\text{pop}},\ j = 1, \ldots, N
\end{aligned} \tag{15}$$

where N_{pop} and N are population size and function dimension, respectively, and U is the uniform distribution operator. Also $[low_j, up_j]$ is variables' predefined interval boundaries.

(2) Selection-I
The BSA calculates the search direction by defining the historical population *oldPOP*. The initial historical population is generated as follows:

$$\begin{aligned}
&oldPOP_{i,j} = U(low_j, up_j)\\
&\quad \text{for } i = 1, \ldots, N_{\text{pop}},\ j = 1, \ldots, N
\end{aligned} \tag{16}$$

In BSA an option is provided to redefine *oldPOP* at the beginning of each iteration through the 'if-then' rule in Eq. (17):

$$\text{if } a < b \text{ then } oldPOP = POP \backslash a, b = U(0, 1) \tag{17}$$

Equation (17) provides a memory for BSA. It ensures that BSA designates a population belonging to a randomly selected previous generation as the historical population and remembers this historical population until it is changed. After *oldPOP* is determined, a randomly change in the order of the individuals in *oldPOP* is performed using Eq. (18)

$$oldPOP = permuting(oldPOP) \tag{18}$$

The permuting function used in Eq. (18) is a random shuffling function.

(3) Mutation
In this stage of algorithm, an initial form of trial population *mutPOP* is generated using Eq. (19)

$$mutPOP = POP + F \cdot (oldPOP - POP) \tag{19}$$

where F is a control parameter, and $(oldPOP - POP)$ can be considered as amplitude of the search direction matrix. This amplitude is controlled using control parameter of F. Because of using the historical population to calculate the search direction matrix, BSA generates a trial population, taking partial advantage of its experiences from previous generations.

In Eq. (19), a trial point is generated by a differential operator. This operator propels the position of current population toward the corresponding members in the historical population. To provide a faster convergence, this research proposes a new operator by inspiring from mutation operators of DE algorithm, to generate initial trial population:

$$\begin{aligned}
&\text{for } i = 1\ to\ N_{\text{pop}}\\
&\quad mutPOP_i = POP_i + F \cdot (POPbest - POP_i)\\
&\quad + F \cdot (oldPOP_i - POP_i)\\
&\text{end for}
\end{aligned} \tag{20}$$

where POP_i and $oldPOP_i$ are ith member of population and historical population, respectively. *POPbest* is the best member of population found so far.

(4) Crossover
After generation of an initial form of trial population *mutPOP*, the final form of the trial population *trialPOP* is generated in the BSA's crossover procedure. Trial individuals with better fitness values for the optimization problem are used to evolve the target population individuals. BSA's crossover process has two steps. The first step calculates a random binary integer-valued matrix (*map*) of size $N_{\text{pop}} \times N$ that indicates the individuals of *trialPOP* to be manipulated by using the relevant individuals of *POP*. Based on this strategy, if $map_{i,j} = 1$, where $i = 1, \ldots, N_{\text{pop}}$ and $j = 1, \ldots, N$, *trialPOP* is updated with $trialPOP_{i,j} = POP_{i,j}$.

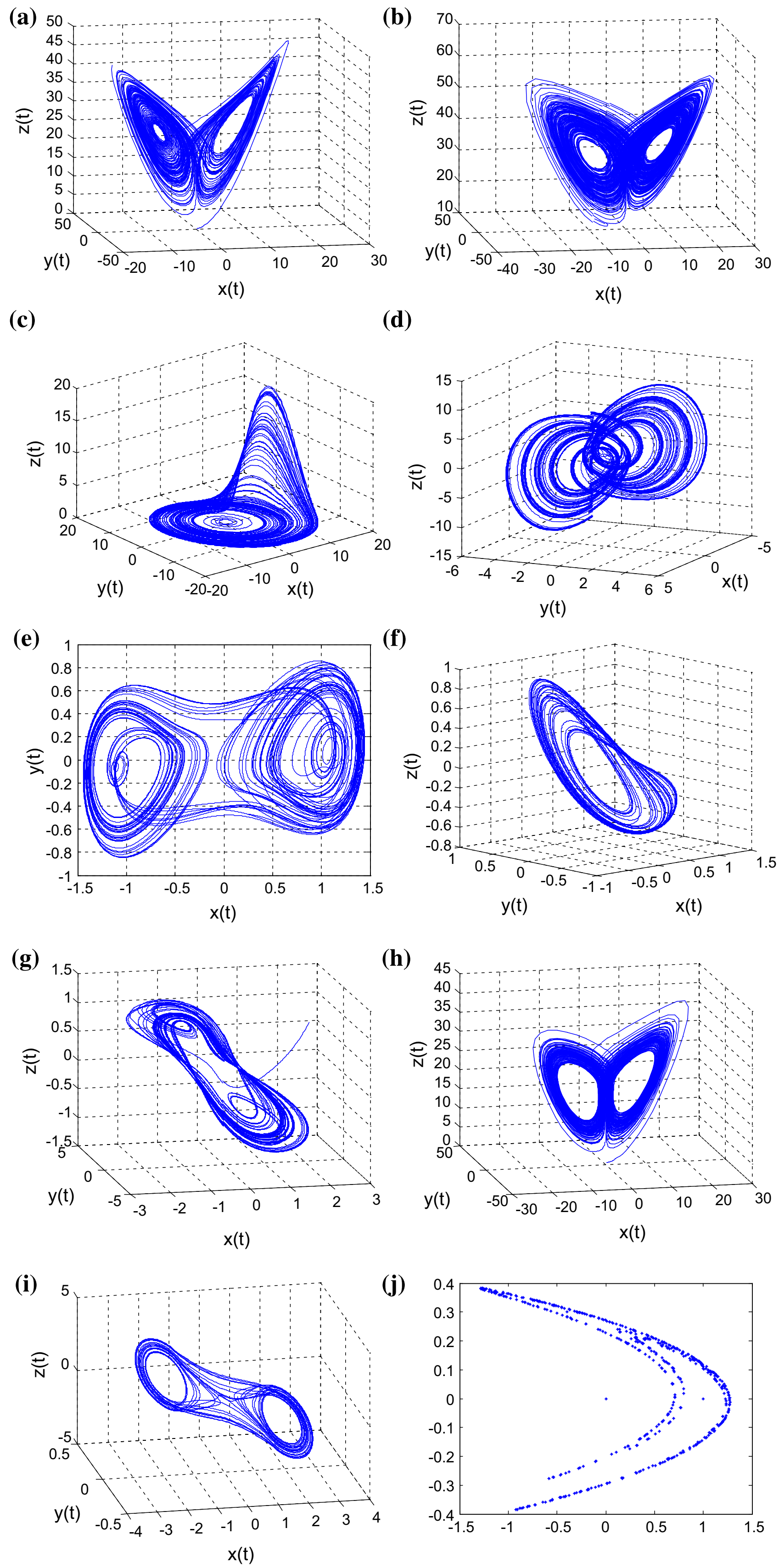

Fig. 1 Phase portraits of chaotic systems: **a** Lorenz, **b** Chen, **c** Rossler, **d** Arneodo, **e** Duffing, **f** Genesio-Tesi, **g** Financial, **h** Lu, **i** Chuas, **j** Henon

Begin SBSA

Step 1: generate and evaluate initial population (POP) of size N_{pop} by Eq. (9).

Step 2: generate and evaluate initial historical population ($oldPOP$) of size N_{pop} by Eq. (10).

Step 3: partition POP in m groups with n members where $N_{pop} = m \times n$ ($Group$).

Step 4: partition $oldPOP$ in m groups with n members where $N_{pop} = m \times n$ ($oldGroup$).

Step 5: apply the original BSA (*Step3.* 0 to *Step* 3.8) to improve each group of POP for $k_{\max}$ iterations as follow:

For $i = 1$ *to* m

Step 3.0:set counter $k = 1$.
Step 3.1: **While** $k \le k_{max}$
Step 3.2: update the current historical group
$if\ \ a < b\ \ then\ \ oldGroup_i = Group_i \ \backslash\ a,b = U(0,1)$
$oldGroup_i = permuting(oldGroup_i)$
Step 3.2: generate the initial trial of current groups i.e. $mutGroup_i$ by Eq. (13).
$mutGroup_i = Group_i + F\,.\,(oldGroup_i - Group_i)$
Step 3.3: calculate a random binary integer-valued matrix of current groups i.e. $mapGroup_i$ of size $n \times N$:
$mapGroup_{i\ 1:n,1:N} = 1$
If $(c < d \ \ \backslash\ c,d = U(0,1))$ **Then**
For $p = 1$ *to* n
$mapGroup_{i\ p,u(1:\lceil rate_{mix}.N.U(0,1)\rceil)} = 0 \ \backslash\ u = permuting(N)$
End For
Else
For $p = 1$ *to* n
$mapGroup_{i\ p,randi(D)} = 0$
End For
End If
Step 3.4: update $trialGroup_i$:
$trialGroup_i = mutGroup_i$
For $p = 1$ *to* n
For $q = 1$ *to* N
If $mapGroup_{i\ p,q} = 1$ **Then**
$trialGroup_{i_{p,q}} = Group_{i_{p,q}}$
End If
End For
End For
Step 3.5: generate the final trial members of current group $trialGroup_i$ by updating infeasible solutions.
Step 3.6: update the members of current group according to a greedy selection:
For $p = 1$ *to* n
If $f(trialGroup_{i_p}) < f(Group_{i_p})$ **Then**
$Group_{i_p} = trialGroup_{i_p}$
End If
End For
Step 3.7: update the global best member $POPbest$ if a better individual is found by $trialGroup_i$.
Step 3.8: set $k = k + 1$.
End While

End For

Step 6: shuffle the population.

Step 7: check the stopping criteria if are not met go to *Step 2*.

End SBSA

Fig. 2 Steps of SBSA

Some individuals of the trial population $trialPOP_{i,j}$ obtained may exceed the lower and upper bounds of search space. Hence, the second step is designed to update these infeasible solutions with randomly generated individuals as in Eq. (15).

(5) Section-II

After generation of final form of candidate members, in the selection-II stage and according to a one-to-one spawning strategy and greedy selection, the individuals in *trialPOP* that have better fitness values are used to update the corresponding individuals in *POP*. So value of cost function in the point $trialPOP_i$, where $i = 1, \ldots, N_{\text{pop}}$, is evaluated, and while $f(trialPOP_i) < f(POP_i)$ POP_i is replaced with $trialPOP_i$; otherwise, no replacement occurs. Moreover, if the best individual of *POP* (*POPbest*) has a better fitness value than the global minimum value obtained so far by BSA, the global minimizer is replaced by *POPbest*, and the global minimum value is replaced by the fitness value of *POPbest*.

4 Shuffled backtracking search optimization

The concept of group search has been used in some EAs such as shuffled complex evolution (SCE) (Duan et al. 1993), SFLA and shuffled DE (SDE) (Ahandani et al. 2010). These algorithms have a same structure. The main trait of them is a grouping search ability to provide some parallel attempts to obtain promising areas based on participating each member in previous experience of all other members. These algorithms have three main stages: partitioning, local search and shuffling. They begin with a population of points distributed randomly throughout the feasible search space. Then in partitioning stage, population is partitioned into several parallel groups. The different groups, which can be perceived as a set of parallel cultures, perform a local search independently using an evolutionary process to continuously evolve their quality for a defined maximum number of iteration. Then in shuffling stage, all evolved complexes are combined together into a single population and the stopping criteria are checked that if are not met the partitioning, local search and shuffling process are continued. The main difference of these algorithms is related their employed evolutionary strategy in local search stage. For example, the SCE, the SFL and the SDE use Nelder–Mead simplex search, PSO and DE, respectively.

This research employs a same approach to improve the original BSA by proposing a shuffled BSA (SBSA). The SBSA has a same structure in comparison with SCE, SFL and SDE algorithms; only the SBSA uses a BSA algorithm to evolve independently member of groups. So the SBSA merges the strengths of original BSA with group search ability. In the SBSA, to make groups on the assumption that partitioning m groups, each containing n members, after sorting of population in a decreasing order in terms of function evaluation, member ranking 1 goes to group 1, member ranking 2 goes to group 2,…, member ranking m goes to group m, then second member of each subset is assigned as: member ranking $(m+1)$ goes to group 1, member ranking $(m+2)$ goes to group 2,…, member ranking $(m+m)$ goes to group m. This process continues to assign all members into groups. Steps of the SBSA with m groups, n members of each group and $k_{\max}$ defined iteration number of evolutionary process are shown in Fig. 2. In this algorithm, $Group_i$ and $oldGroup_i$ denote ith group and historical group, respectively. $mutGroup_i$ is trial members of ith group, $mapGroup_i$ is binary integer-valued matrix of ith group, and *POPbest* is the best member found so far. Also a, b, c and d are random numbers with uniform distribution selected randomly in [0,1].

Table 1 Statistical results of different versions of BSA on Lorenz chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−27)

Parameters		θ_1	θ_2	θ_3
True values		10	28	2.6667
Algorithms				
SBSA1	Best	10	28	2.6667
	Mean	10	28	2.6667
	Variance	0	0	4.6811e−16
	Success Rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	**0**		
	Variance-SSE	**0**		
SBSA2	Best	10	28	2.6667
	Mean	10	28	2.6667
	Variance	0	0	4.6811e−16
	Success Rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	**0**		
	Variance-SSE	**0**		
BSA1	Best	10	28	2.6667
	Mean	10	28	2.6667
	Variance	0	0	4.6811e−16
	Success Rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	**0**		
	Variance-SSE	**0**		
BSA2	Best	10	28	2.6667
	Mean	10	28	2.6667
	Variance	0	0	4.6811e−16
	Success Rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	**0**		
	Variance-SSE	**0**		

Table 2 Statistical results of different versions of BSA on the literature on Chen chaotic system

Parameters		θ_1	θ_2	θ_3	θ_4
True values		35	3	28	−7
Algorithms					
SBSA1	Best	35	3	28	−7
	Mean	35.002	3	27.992	−7.0076
	Variance	0.0054283	4.6114e−05	0.024826	0.024135
	Success rate	**100**			
	Best-SSE	8.0745e−25			
	Mean-SSE	0.0006894			
	Variance-SSE	0.0021802			
SBSA2	Best	35	3	28	−7
	Mean	35	3	28	−7
	Variance	2.0202e−13	5.9617e−14	1.0681e−12	7.2049e−13
	Success rate	**100**			
	Best-SSE	**3.0124e−26**			
	Mean-SSE	**1.1889e−22**			
	Variance-SSE	**3.7375e−22**			
BSA1	Best	35	3	28	−7
	Mean	35.017	2.9999	27.926	−7.0723
	Variance	0.05223	0.00041512	0.23559	0.22863
	Success rate	84			
	Best-SSE	3.2083e−19			
	Mean-SSE	0.064089			
	Variance-SSE	0.20267			
BSA2	Best	35	3	28	−7
	Mean	35	3	28	−7
	Variance	6.7258e−12	1.4295e−12	3.093e−10	3.7019e−10
	Success rate	**100**			
	Best-SSE	3.0322e−26			
	Mean-SSE	1.8722e−18			
	Variance-SSE	5.915e−18			

5 Computational results

In this section, different experiments are carried out to assess the performance of proposed algorithm. These experiments are designed to identify or estimate parameters of a wide variety of chaotic systems. A plenty of experiments were performed to give sufficiently good results for different problems, and the obtained values to set parameters of the BSA and the SBSA are as follows: $N_{\text{pop}} = 24$, $m = 4$, $n = 6$ and $k_{\max} = 10$. Also control parameters of BSA, i.e., F is considered a random number in the range of [0,2] and $rate_{mix}$ is considered equal to 1 as proposed in Civicioglu (2013).

Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 and 18 show how good our proposed algorithms are. Three different versions of BSA are proposed here and are compared against original BSA. In all comparisons, the original BSA which uses Eq. (13) to generate initial trial point is called here BSA1, and the original BSA which uses Eq. (14) to generate initial trial point is called here BSA2. Also the proposed SBSA with Eq. (13) to generate initial trial point is called here SBSA1 and the proposed SBSA with Eq. (14) to generate initial trial point is called here SBSA2. In addition, typical simulation results (including the convergent processes of objective value) are presented for the different chaotic systems with Figs. 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. For all problems, the statistical simulation results of 50 independent runs are reported, where “Best”, “Mean” and “Variance” denote values of unknown parameters in the best run, mean values of unknown parameters in all runs and the variance of values of unknown parameters, respectively. Also “Best-SSE”, “Mean-SSE” and “Variance-SSE” denote the best found SSE value, mean of the best SSE values and variance of found SSE values in all independent runs, respectively. Also “Success Rate” denotes percent of successful

Table 3 Statistical results of different versions of BSA on Rossler chaotic system (*Note* Zero value on this example means a number smaller than 1.0E−33)

Parameters		θ_1	θ_2	θ_3
True values		0.5	0.2	10
Algorithms				
SBSA1	Best	0.5	0.2	10
	Mean	0.5	0.2	10
	Variance	3.5532e−13	1.8272e−11	5.5414e−10
	Success rate	**100**		
	Best-SSE	1.5869e−26		
	Mean-SSE	5.0957e−21		
	Variance-SSE	1.4356e−20		
SBSA2	Best	0.5	0.2	10
	Mean	0.5	0.2	10
	Variance	**2.8184e−16**	**4.7225e−14**	**8.1344e−13**
	Success rate	**100**		
	Best-SSE	6.7878e−31		
	Mean-SSE	**1.419e−27**		
	Variance-SSE	**4.2138e−27**		
BSA1	Best	0.5	0.2	10
	Mean	0.5	0.20004	10.002
	Variance	2.678e−06	0.00013607	0.0062735
	Success rate	**100**		
	Best-SSE	7.4702e−32		
	Mean-SSE	5.2424e−08		
	Variance-SSE	1.6578e−07		
BSA2	Best	0.5	0.2	10
	Mean	0.5	0.2	10
	Variance	2.2524e−15	1.5029e−13	5.0567e−12
	Success rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	6.4163e−26		
	Variance-SSE	1.3754e−25		

runs. In this research, a run is called successful if its corresponding SSE reaches less than 0.01. Also a fixed value of function evaluations equal to 5000 $\times$ N is considered as termination criterion. The best results in tables are presented in a bold face.

5.1 Simulation about Lorenz system

Table 1 shows the statistical results of different methods for the Lorenz chaotic system. The obtained results of these tables show that all algorithms are consistent and they converge to the true values as accurately as possible. Figure 3 shows the convergence of J corresponding to a typical run for the Lorenz chaotic system. An interesting observation from these curves is that all algorithms reach to vicinity of true values of parameters in early number of function evaluations.

5.2 Simulation about Chen system

Table 2 shows the statistical results of different methods for the Chen chaotic system. It can be seen that the SBSA2 outperforms other algorithms in terms of all considered aspects. The SBSA2 obtains the best minimum and mean of SSE values. The second-best results are related to the BSA2. The SBSA1 beside two aforementioned algorithms obtains a success rate of 100. These results demonstrate that both of the proposed ideas to improve BSA, i.e., grouping search and new operator, to generate initial trial point are effective to evolve the original BSA. Comparatively, the performances of original BAS, i.e., BSA1, are the worst among the four algorithms. Figure 4 shows the convergence of J corresponding to a typical run for the Chen chaotic system. From these curves, it is clear that the BSA2 and SBSA2 can achieve the best results with a high solution accuracy. So results of Table 2 and Fig. 4 demonstrate better effectiveness and

Table 4 Statistical results of different versions of BSA on Arneodo chaotic system

Parameters		θ_1	θ_2	θ_3	θ_4
True values		−5.5	3.5	0.8	−1
Algorithms					
SBSA1	Best	−5.5	3.5	0.8	−1
	Mean	−5.5007	3.5005	0.79957	−1.0009
	Variance	0.0021175	0.0011326	0.0005653	0.0013613
	Success rate	**100**			
	Best-SSE	9.7794e−12			
	Mean-SSE	0.00068345			
	Variance-SSE	0.0012983			
SBSA2	Best	−5.5	3.5	0.8	−1
	Mean	−5.5	3.5	0.8	−1
	Variance	2.1781e−06	7.4556e−07	4.1555e−07	4.367e−07
	Success rate	**100**			
	Best-SSE	1.7089e−19			
	Mean-SSE	2.8141e−10			
	Variance-SSE	8.0836e−10			
BSA1	Best	−5.5	3.5	0.8	−1
	Mean	−5.4998	3.4999	0.80001	−0.99999
	Variance	0.0010401	0.0003158	0.00029165	0.00042342
	Success rate	**100**			
	Best-SSE	9.258e−11			
	Mean-SSE	0.0002979			
	Variance-SSE	0.00083936			
BSA2	Best	−5.5	3.5	0.8	−1
	Mean	−5.5	3.5	0.8	−1
	Variance	5.9728e−09	2.8232e−09	2.1612e−10	7.7548e−10
	Success rate	**100**			
	Best-SSE	**3.7001e−24**			
	Mean-SSE	**2.3124e−15**			
	Variance-SSE	**6.8932e−15**			

robustness of the BSA2 and SBSA2 than the other two algorithms for parameter identification of the Chen chaotic system.

5.3 Simulation about Rossler system

Table 3 shows the statistical results of different methods for the Rossler chaotic system. Based on results of this table, all algorithms obtain a complete success rate. Also it can be seen that the BSA2 has the best performance in terms of Best-SSE and the SBSA2 obtains the best Mean-SSE. Also the BSA1 obtains again the worst Mean-SSE value among all algorithms. Figure 5 shows the convergence of J corresponding to a typical run for the Rossler chaotic system. The obtained curves of Fig. 5 show the fast convergence speed of algorithms. So results of Table 3 and Fig. 5 demonstrate better effectiveness and robustness of the BSA2 and SBSA2 than the other two algorithms for parameter identification of the Rossler chaotic system.

5.4 Simulation about Arneodo system

Table 4 shows the statistical results of different methods for the Arneodo chaotic system. As obtained in previous tables, the BSA2 and SBSA2 are again two superior algorithms on this system. The BSA1 obtains the best results, and the SBSA2 has the second-best performance. The SBSA1 has a better Best-SSE than the BSA1, but the BSA1 obtains a less Mean-SSE than the SBSA1. Figure 6 shows the convergence of J corresponding to a typical run for the Arneodo chaotic system. These curves show that the BSA2 and SBSA2 have faster convergence speed than the other two algorithms. So results of Table 4 and Fig. 6 demonstrate better effectiveness and robustness of the BSA2 and SBSA2 than the other

Table 5 Statistical results of different versions of BSA on Duffing chaotic system

Parameters		θ_1	θ_2	θ_3
True values		0.15	0.31	1
Algorithms				
SBSA1	Best	0.15	0.31	1
	Mean	0.16349	0.31125	0.99455
	Variance	0.027409	0.022562	0.010724
	Success rate	90		
	Best-SSE	2.389e−18		
	Mean-SSE	0.010944		
	Variance-SSE	0.029847		
SBSA2	Best	0.15	0.31	1
	Mean	0.15	0.31	1
	Variance	1.6775e−10	1.2979e−10	1.5822e−10
	Success rate	**100**		
	Best-SSE	**3.1228e−22**		
	Mean-SSE	**1.5444e−17**		
	Variance-SSE	**4.7204e−17**		
BSA1	Best	0.15	0.31	1
	Mean	0.21784	0.3565	0.97842
	Variance	0.09172	0.077196	0.023108
	Success rate	50		
	Best-SSE	1.006e−13		
	Mean-SSE	0.13291		
	Variance-SSE	0.27939		
BSA2	Best	0.15	0.31	1
	Mean	0.15	0.31	1
	Variance	5.2031e−07	4.0743e−07	3.2014e−07
	Success rate	**100**		
	Best-SSE	2.8594e−21		
	Mean-SSE	1.999e−11		
	Variance-SSE	6.3215e−11		

two algorithms for parameter identification of the Arneodo chaotic system.

5.5 Simulation about Duffing system

Table 5 shows the statistical results of different methods for the Duffing chaotic system. It can be seen that all the results got by SBSA2 are consistent and it has the best performance. The second-best performance is related to the BSA2. Among all algorithms, only the SBSA2 and BSA2 have a success rate of 100. Also the original BSA1 is the worst among the four algorithms. Figure 7 shows the convergence of J corresponding to a typical run for the Duffing chaotic system. These curves confirm the obtained results of Table 5 about a better performance of the proposed algorithms in this research. The SBSA2 converges quickly and detects optimums in a few iterations.

5.6 Simulation about Genesio-Tesi system

Table 6 shows the statistical results of different methods for the Genesio-Tesi chaotic system. As obtained from aforementioned experiments, it is seen from the results of this table that the BSA2 and SBSA2 outperform two other algorithms. The BSA2 and SBSA2 have a complete success rate. The SBSA2 obtains the best Best-SSE, and the BSA2 has the best Mean-SSE. Also the SBSA1 has a better performance than the original BSA1. Figure 8 shows the convergence of J corresponding to a typical run for the Genesio-Tesi chaotic system. The trajectories of J during the evolutionary procedure confirm a better performance of the BSA2 and SBSA2. They converge quickly and detect optimums in a few iterations.

Table 6 Statistical results of different versions of BSA on Genesio-Tesi chaotic system

Parameters		θ_1	θ_2	θ_3	θ_4
True values		1.1	1.1	0.45	1
Algorithms					
SBSA1	Best	1.1	1.1	0.45	1
	Mean	1.1205	1.0966	0.46444	1.0115
	Variance	0.034466	0.0056023	0.024154	0.019434
	Success rate	90			
	Best-SSE	1.3706e−12			
	Mean-SSE	0.0026245			
	Variance-SSE	0.0054221			
SBSA2	Best	1.1	1.1	0.45	1
	Mean	1.1	1.1	0.45	1
	Variance	5.1972e−10	8.4846e−11	3.4375e−10	2.1706e−10
	Success rate	**100**			
	Best-SSE	**1.7014e−22**			
	Mean-SSE	3.8667e−18			
	Variance-SSE	7.6096e−18			
BSA1	Best	1.1	1.1	0.45	1
	Mean	1.1664	1.0898	0.49623	1.0382
	Variance	0.10333	0.015727	0.071903	0.059669
	Success rate	80			
	Best-SSE	1.1635e−11			
	Mean-SSE	0.020595			
	Variance-SSE	0.043191			
BSA2	Best	1.1	1.1	0.45	1
	Mean	1.1	1.1	0.45	1
	Variance	8.9541e−11	1.7009e−11	6.2485e−11	4.1695e−11
	Success rate	**100**			
	Best-SSE	2.4288e−22			
	Mean-SSE	**4.437e−20**			
	Variance-SSE	**4.614e−20**			

5.7 Simulation about financial system

Table 7 shows the statistical results of different methods for the Financial chaotic system. The obtained results on this system make confidence about superior performance of the SBSA2 and BSA2 methods. Among four algorithms, only the two algorithms converge to optimum point as accurately as possible in all runs. After them, the SBSA1 has the second-best results and the BSA1 is has the worst performance; however, it has a success rate of 100. Figure 9 shows the convergence of J corresponding to a typical run for the Financial chaotic system. From Fig. 9, it is clear that the all algorithms have a quick convergence toward optimum value.

5.8 Simulation about Lu system

Table 8 shows the statistical results of different methods for the Lu chaotic system. According to obtained results on this chaotic system, only the BSA1 cannot achieve a success rate of 100. However, the BSA1 beside BSA2 obtains the best Best-SSE value. Also the SBSA2 has the best Mean-SSE, and the BSA1 has the worst Mean-SSE. Figure 10 shows the convergence of J corresponding to a typical run for the Lu chaotic system. These curves depict the evolving processes of fitness value for all the algorithms. It is clear that the all algorithms have a quick convergence toward optimum value.

5.9 Simulation about Chuas system

Table 9 shows the statistical results of different methods for the Chuas chaotic system. This system has five unknown parameters to be estimated, and the obtained results show that it is very difficult to be solved. Table 9 highlights the efficiency and robustness of the proposed algorithms here. The SBSA2 and BSA2 have a considerably better performance in terms of all considered aspects than two another algorithms.

Table 7 Statistical results of different versions of BSA on Financial chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−30)

Parameters		θ_1	θ_2	θ_3
True values		1	0.1	1
Algorithms				
SBSA1	Best	1	0.1	1
	Mean	1	0.1	1
	Variance	3.4716e−16	1.2903e−16	5.7807e−16
	Success rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	2.6649e−29		
	Variance-SSE	4.7988e−29		
SBSA2	Best	1	0.1	1
	Mean	1	0.1	1
	Variance	0	1.4628e−17	0
	Success rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	**0**		
	Variance-SSE	**0**		
BSA1	Best	1	0.1	1
	Mean	1	0.1	1
	Variance	5.9257e−10	1.97e−10	7.2292e−10
	Success rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	6.7651e−17		
	Variance-SSE	2.1393e−16		
BSA2	Best	1	0.1	1
	Mean	1	0.1	1
	Variance	0	1.4628e−17	0
	Success rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	**0**		
	Variance-SSE	**0**		

The SBSA1 and BSA1 cannot obtain any success rate on this system. Results of this table demonstrate that the proposed operator to generate initial trial point is very effective to evolve performance of BSA and SBSA. Figure 11 shows the convergence of J corresponding to a typical run for the Chuas chaotic system. It is clear that the SBSA2 and BSA2 can achieve good results, whereas the SBSA1 and BSA1 get stuck on local minimums.

5.10 Simulation about Henon system

Table 10 shows the statistical results of different methods for the Henon chaotic system. This system has two unknown parameters to be estimated, and it is not difficult to be solved. All algorithms have a success rate of 100, and among all algorithms, the SBSA2 converges to optimum point as accurately as possible in all runs. Figure 12 shows the convergence of J corresponding to a typical run for the Henon chaotic system. From Fig. 12, it is clear that the all algorithms have a quick convergence toward optimum value.

To provide an overall consequence from Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 about performance of algorithms on aforementioned 10 chaotic systems, Tables 11 and 12 show rank of each algorithm based on Best-SSE and Mean-SSE for the algorithms over the set of 10 chaotic systems, respectively. According to results of Tables 11 and 12, the SBSA2 obtains rank 1 on 7 and 8 problems based on Best-SSE and Mean-SSE, respectively. Also the BSA2 obtains rank 1 on 6 and 4 problems based on Best-SSE and Mean-SSE, respectively. It also obtains rank 2 on 4 and 6 other problems based on Best-SSE and Mean-SSE, respectively. The SBSA1 obtains rank 1 on 3 and 1 problems based on Best-SSE and Mean-SSE, respectively. It also obtains rank 3 on 3 and 8 other problems based on Best-SSE and Mean-SSE, respectively. The

Table 8 Statistical results of different versions of BSA on Lu chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−30)

Parameters		θ_1	θ_2	θ_3
True values		36	3	20
Algorithms				
SBSA1	Best	36	3	20
	Mean	36	3	20
	Variance	0.00090994	8.4618e−05	0.0006192
	Success rate	**100**		
	Best-SSE	1.1402e−23		
	Mean-SSE	0.00022481		
	Variance-SSE	0.00071091		
SBSA2	Best	36	3	20
	Mean	36	3	20
	Variance	2.2117e−13	4.0723e−14	2.1932e−14
	Success rate	**100**		
	Best-SSE	6.2283e−27		
	Mean-SSE	**3.2629e−24**		
	Variance-SSE	**8.7568e−24**		
BSA1	Best	36	3	20
	Mean	36.006	2.9991	20
	Variance	0.019967	0.0028872	7.5949e−05
	Success rate	94		
	Best-SSE	0		
	Mean-SSE	0.0088809		
	Variance-SSE	0.027977		
BSA2	Best	36	3	20
	Mean	36	3	20
	Variance	3.2522e−13	7.9114e−14	6.374e−14
	Success rate	**100**		
	Best-SSE	**0**		
	Mean-SSE	1.1329e−23		
	Variance-SSE	3.2446e−23		

BSA1 obtains rank 1 on 4 and 1 problems based on Best-SSE and Mean-SSE, respectively. It also obtains rank 4 on 3 and 8 other problems based on Best-SSE and Mean-SSE, respectively.

A nonparametric analysis over the obtained results from Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 using the Wilcoxon signed-rank test with $\alpha = 0.05$ in terms of best run (Best-SSE) and average of all runs (Mean-SSE) is provided in Table 13.

Pairwise comparisons of Table 13 show that the BSA2 has a significant difference with respect to the BSA1 and SBSA1 in terms of Best-SSE. Also the SBSA2 considerably outperforms the BSA1 in terms of Best-SSE. Also difference of BSA2 and SBSA2 based on this test is not significant; however, this test confirms a better performance of SBSA2. Also based on Mean-SSE, BSA2 and SBSA2 have a significantly better performance than two other algorithms. Also this test confirms a better Mean-SSE performance of the SBSA2 than the BSA2; however, this difference is not significant.

Table 4 compares algorithms over the set of 10 chaotic systems based on average of running time.

Based on results of time comparison, since all algorithms have a same stopping criterion, i.e., a fixed value of function evaluation, there is not a significantly difference among running time of them. Also Table 14 clearly shows that the proposed ideas in this study to evolve the performance of original BSA are not more time-consuming than the original structure or operators of BSA.

From Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14 and Figs. 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12, the following results are observed: (i) both of the proposed ideas to improve performance of original BSA, i.e., group search ability and new operator, to generate initial trial population are effective.

Table 9 Statistical results of different versions of BSA on Chuas chaotic system

Parameters		θ_1	θ_2	θ_3	θ_4	θ_5
True values		10.725	10.593	−0.268	−0.7872	1.1726
Algorithms						
SBSA1	Best	10.451	10.489	−0.23641	−0.78419	1.192
	Mean	10.144	9.7673	−0.24409	−0.74858	1.2098
	Variance	0.61133	0.67134	0.07469	0.036182	0.050817
	Success rate	0				
	Best-SSE	0.87688				
	Mean-SSE	30.965				
	Variance-SSE	34.998				
SBSA2	Best	10.725	10.593	−0.268	−0.7872	1.1726
	Mean	10.726	10.593	−0.26819	−0.7872	1.1726
	Variance	0.0066277	0.0059195	0.0005676	0.0002169	8.7178e−05
	Success rate	**92**				
	Best-SSE	**6.3591e−11**				
	Mean-SSE	**0.0064617**				
	Variance-SSE	**0.019529**				
BSA1	Best	10.496	10.469	−0.24483	−0.783	1.1866
	Mean	10.27	9.5329	−0.27958	−0.73533	1.2013
	Variance	0.67125	0.76849	0.077709	0.039163	0.034607
	Success rate	0				
	Best-SSE	0.59619				
	Mean-SSE	48.819				
	Variance-SSE	52.064				
BSA2	Best	10.725	10.593	−0.268	−0.7872	1.1726
	Mean	10.725	10.593	−0.268	−0.7872	1.1726
	Variance	0.08002	0.023131	0.016243	0.0020604	0.00017805
	Success rate	88				
	Best-SSE	6.774e−09				
	Mean-SSE	0.19118				
	Variance-SSE	0.60419				

(ii) A considerably better performance of the SBSA2 and BSA2 based on Mean-SSE demonstrates that the proposed ideas can evolve robustness and consistence of algorithm. (iii) Inserting a new operator to generate initial trial population considerably makes better exploitation ability of original BSA algorithm so that the BSA2 could find solutions with a high accuracy. (iv) By providing a group search ability for the BSA algorithm and presenting the SBSA, the exploration ability of original BSA was improved. Also by inserting a new operator to generate initial trial population, exploitation ability of SBSA was evolved. So by combining these two concepts, we can see that the SBSA2 has the best performance among four algorithms.

5.11 Comparison with results reported in the literature

In this section, a comparison between the proposed algorithms in this research, i.e., BSA2 and SBSA2, and some other algorithms reported in the different literature to solve parameter identification of chaotic systems is performed. Ho et al. (2010) proposed an improved DE algorithm, named the Taguchi-sliding-based DE algorithm (TSBDEA), to solve the problem of parameter identification for Chen, Lü and Rossler chaotic systems. The TSBDEA combines the DE with the Taguchi-sliding-level method (TSLM). The TSLM is used as the crossover operation of the DEA. Then, the systematic reasoning ability of the TSLM is provided to select the better offspring to achieve the crossover and consequently enhance the DE. Based on Ho et al. (2010), sampling time is 0.0005 s for Chen and Lü systems, the sampling time is 0.001 s for Rossler system, and the total sampling number L is 100 for all three systems. Also number of function evaluations for Chen and Lü systems is considered equal to 10,000, and for the Rossler system, it is considered equal to 12,000. They used mean square error (MSE) as cost function as follows:

Table 10 Statistical results of different versions of BSA on Henon chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−28)

Parameters		θ_1	θ_2
True values		36	3
Algorithms			
SBSA1	Best	36	3
	Mean	36	3
	Variance	6.367e−16	9.6148e−17
	Success rate	**100**	
	Best-SSE	**0**	
	Mean-SSE	6.1775e−29	
	Variance-SSE	1.0005e−28	
SBSA2	Best	36	3
	Mean	36	3
	Variance	2.3406e−16	5.8514e−17
	Success rate	**100**	
	Best-SSE	**0**	
	Mean-SSE	**0**	
	Variance-SSE	**0**	
BSA1	Best	36	3
	Mean	36	3
	Variance	6.6201e−16	1.1556e−16
	Success rate	**100**	
	Best-SSE	**0**	
	Mean-SSE	8.2562e−29	
	Variance-SSE	1.1164e−28	
BSA2	Best	36	3
	Mean	36	3
	Variance	6.3238e−16	7.4015e−17
	Success rate	**100**	
	Best-SSE	**0**	
	Mean-SSE	4.2015e−29	
	Variance-SSE	8.959e−29	

Table 11 Ranks based on Best-SSE for the algorithms over the set of 10 chaotic systems

Algorithms	BSA1	BSA2	SBAS1	SBAS2
Lorenz	1	1	1	1
Chen	4	2	3	1
Rossler	2	1	4	3
Arneodo	4	1	3	2
Duffing	4	2	3	1
Genesio-Tesi	3	2	4	1
Financial	1	1	1	1
Lu	1	1	4	3
Chuas	3	2	4	1
Henon	1	1	1	1
Sum of ranks	24	14	28	15

Table 12 Ranks based on Mean-SSE for the algorithms over the set of 10 chaotic systems

Algorithms	BSA1	BSA2	SBAS1	SBAS2
Lorenz	1	1	1	1
Chen	4	2	3	1
Rossler	4	2	3	1
Arneodo	3	1	4	2
Duffing	4	2	3	1
Genesio-Tesi	4	1	3	2
Financial	4	1	3	1
Lu	4	2	3	1
Chuas	4	2	3	1
Henon	4	2	3	1
Sum of ranks	28	16	26	12

Table 13 Wilcoxon test applied over the obtained results of Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 in terms of the best run (Best-SSE) and average of all runs (Mean-SSE)

Comparison	Best-SSE			Mean-SSE		
	R^+	R^-	p value	R^+	R^-	p value
BSA1–SBSA2	43	12	0.063	54.5	0.5	**0.008**
BSA2–SBSA2	33	22	0.893	37.5	17.5	0.401
SBSA1–SBSA2	52	3	**0.018**	54.5	0.5	**0.008**
BSA1–BSA2	50	5	**0.028**	54.5	0.5	**0.008**
SBSA1–BSA2	52	3	**0.018**	54.5	0.5	**0.008**
BSA1–SBSA1	24	31	0.735	49.5	5.5	**0.028**

Table 14 A comparison for the algorithms over the set of 10 chaotic systems based on average of running time (*Note* Time values in this table are in terms of seconds)

Algorithms	BSA1	BSA2	SBAS1	SBAS2
Lorenz	24.038	22.629	24.76	27.004
Chen	41.174	44.242	47.191	51.52
Rossler	23.29	24.342	24.811	22.147
Arneodo	35.079	31.689	31.444	31.479
Duffing	18.352	19.215	18.017	18.498
Genesio-Tesi	23.853	21.347	31.131	33.131
Financial	21.413	16.788	18.81	23.034
Lu	25.746	24.63	24.061	23.935
Chuas	35.691	30.522	38.814	38.595
Henon	33.792	33.606	32.132	35.465
Sum of ranks	26	22	24	28

$$\min J = \frac{1}{L}\sum_{k=1}^{L}\left\|x(k) - \hat{x}(k)\right\|^2 \tag{21}$$

Values of parameters for BSA2 and SBSA2 are same to those of considered in aforementioned experiments; only

Table 15 Comparison among results of BSA2 and SBSA2 and results of Ho et al. (2010) on Chen chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−30)

Parameters		θ_1	θ_2	θ_3
True values		36	3	20
Algorithms				
SBSA2	Best	36	3	28
	Mean	36	3	28
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
BSA2	Best	36	3	28
	Mean	36	3	28
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
TSBDEA	Best	36	3	28
	Mean	36	3	28
	Variance	0	0	0
	Best-SSE	0		
	Mean-SSE	0		
	Variance-SSE	0		
DE	Best	35.6099	3.0129	28.2927
	Mean	36.1181	3.0584	28.5292
	Variance	1.0533	0.0518	0.4974
	Best-SSE	0.0441		
	Mean-SSE	0.3415		
	Variance-SSE	0.2765		

Table 16 Comparison among results of BSA2 and SBSA2 and results of Ho et al. (2010) on Lu chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−30)

Parameters		θ_1	θ_2	θ_3
True values		36	3	20
Algorithms				
SBSA2	Best	36	3	20
	Mean	36	3	20
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
BSA2	Best	36	3	20
	Mean	36	3	20
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
TSBDEA	Best	36	3	20
	Mean	36	3	20
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
DE	Best	36.0388	2.9927	20.0006
	Mean	35.9577	3.0015	19.9986
	Variance	0.3650	0.0256	0.0089
	Best-MSE	0.0007		
	Mean-MSE	0.0179		
	Variance-MSE	0.0167		

to provide a better comparison, size of initial population is considered equal to 16. Tables 15, 16 and 17 show this comparison. The obtained results demonstrate that the BSA2, SBSA2 and TSBDEA obtain the best results. They converge to optimal solution in all trial runs. Thus, it can be concluded that the BSA2, SBSA2 and TSBDEA can give a more effective and robust way for estimating the true parameters than the DE.

Tang et al. (2012) applied the DE to search the optimal parameters of commensurate fractional-order chaotic systems when orders are known and unknown. They applied the DE on Lu and Volta chaotic systems and compared the obtained results with GA. Sampling time is 0.005 and 0.0005 s for Lü and Volta systems, respectively, and the total sampling number, L, is 100 for both of them. Also number of function evaluations for these systems is considered equal to 10,000. Tables 18 and 19 compare results of BSA2 and SBSA2 in respect to results reported in Tang et al. (2012) when orders of chaotic systems are known. From these tables, it can be seen that the BSA2 and SBSA2 obtained the best results with respect to the DE and GA so that their estimated parameters are accurately same as the true parameter values, but there exist certain errors between the average results of DE and the true parameter values. Also, none of the results obtained by GA are the same as the true parameter values.

Hu et al. (2015) proposed a hybrid artificial bee colony (HABC) algorithm for identification of uncertain fractional-order chaotic systems. Fractional-order Economic and Rössler chaotic systems are selected to test the performance. Instead of MSE in Eq. (21), the following objective function is defined in Hu et al. (2015) which is a sum square error (SSE) function:

$$\min J = \sum_{k=1}^{N} \left\| X(k) - \hat{X}(k) \right\|^2 \tag{22}$$

To calculate the objective function, the number of samples is set as 300 and the step size is 0.01. The parameters

Table 17 Comparison among results of BSA2 and SBSA2 and results of Ho et al. (2010) on Rossler chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−33)

Parameters		θ_1	θ_2	θ_3
True values		36	3	20
Algorithms				
SBSA2	Best	36	3	20
	Mean	36	3	20
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
BSA2	Best	36	3	20
	Mean	36	3	20
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
TSBDEA	Best	36	3	20
	Mean	36	3	20
	Variance	0	0	0
	Best-MSE	0		
	Mean-MSE	0		
	Variance-MSE	0		
DE	Best	0.1942	0.4052	5.8180
	Mean	0.1999	0.3728	5.7994
	Variance	0.0096	0.2887	0.1651
	Best-MSE	0.0001		
	Mean-MSE	0.0004		
	Variance-MSE	0.0003		

Table 18 Comparison among results of BSA2 and SBSA2 and results of Tang et al. (2012) on Lu chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−30)

Parameters		θ_1	θ_2	θ_3
True values		36	3	20
Algorithms				
SBSA2	Best	25	3	28
	Mean	25	3	28
	Best-MSE	0		
	Mean-MSE	5.6938e−29		
BSA2	Best	25	3	28
	Mean	25	3	28
	Best-MSE	0		
	Mean-MSE	2.1822e−28		
DE	Best	25	3	28
	Mean	25	3	28
	Best-MSE	1.38e−26		
	Mean-MSE	2.18e−7		
GA	Best	25	3	28
	Mean	25.0012	2.7769	27.9071
	Best-MSE	6.51e−5		
	Mean-MSE	12.183		

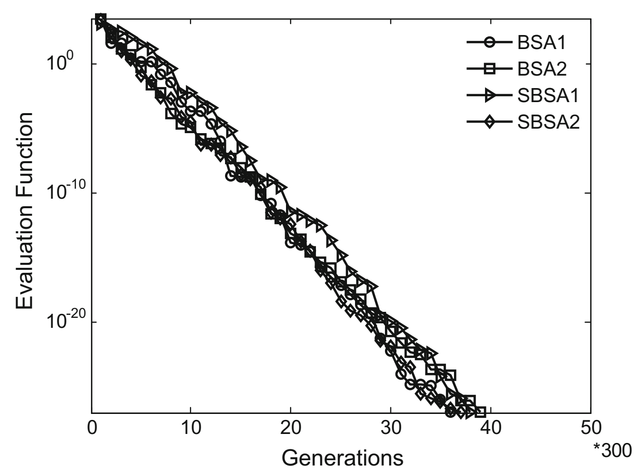

Fig. 3 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Lorenz chaotic system

of HABC algorithm are set as follows: Population size is 100, maximum cycle number of iterations are set as 50 for fractional-order economic chaotic system and 100 for fractional-order Rössler chaotic systems, respectively, the control parameter (*limit*) is 15, and the maximum number of chaotic iteration $N = 300$. The algorithm is executed 15 times for each example, and all runs are terminated after the predefined maximum cycle number of iterations is reached.

In the first part of comparison study in Hu et al. (2015), the HABC was compared against two other versions of ABC, i.e., GABC and EABC on fractional-order Economic chaotic system. Table 20 shows the obtained results of the SBSA2 and the BSA2 against results of aforementioned algorithms. From the simulations results of the fractional-order Economic system, it can be concluded that the BSA2 outperforms all other algorithms based on all considered aspects. Also SBSA2 has a better performance than the GABC, the EABC and the HABC in terms of all criteria, except in comparison with the HABC in terms of Best-SSE.

In the second part of comparison study in Hu et al. (2015), the HABC was compared against three other EAs, i.e., GA, DE and PSO on fractional-order Rössler chaotic system. Table 21 shows the obtained results of the SBSA2 and the BSA2 against results of aforementioned algorithms. Table 21 clearly confirms a considerable better performance of the SBSA2 and the BSA2 in comparison with HABC, GA, DE and PSO. In this table, the BSA2 has the best performance in terms of all considered aspects and the SBSA2 obtains the second-best performance.

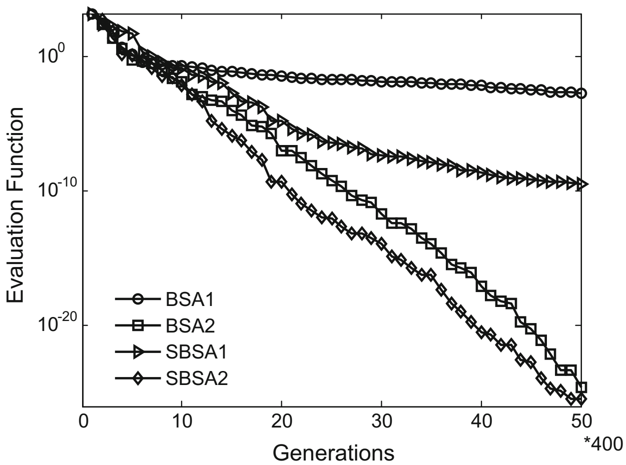

Fig. 4 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Chen chaotic system

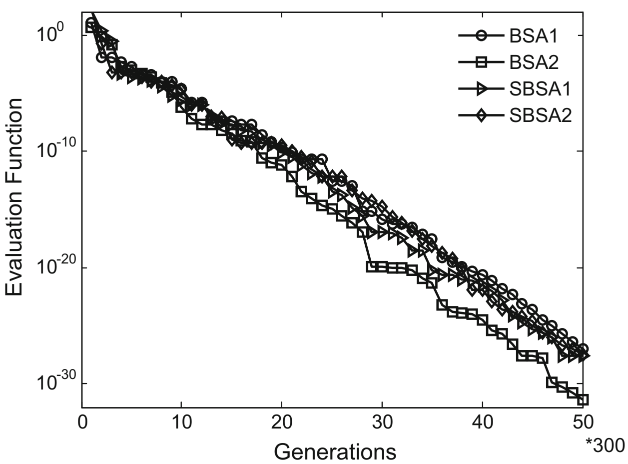

Fig. 5 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Rossler chaotic system

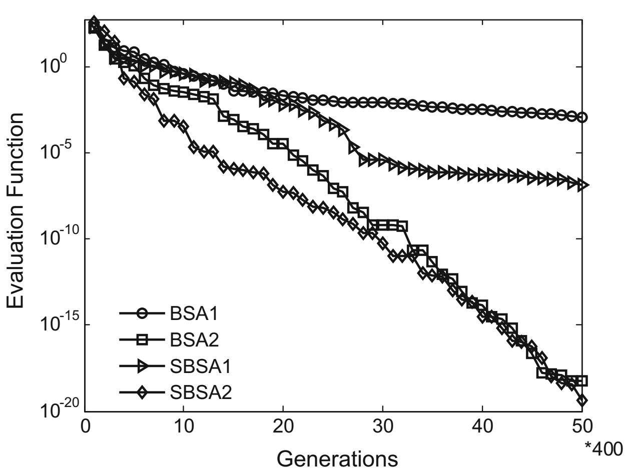

Fig. 6 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Arneodo chaotic system

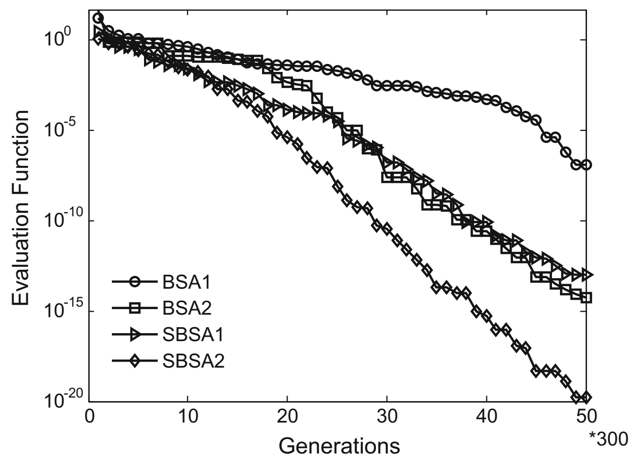

Fig. 7 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Duffing chaotic system

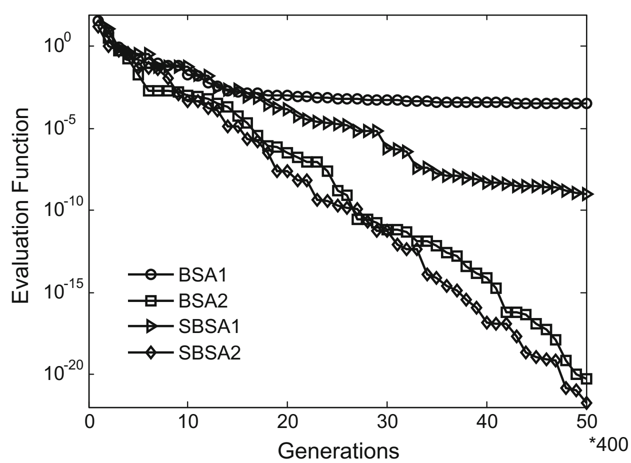

Fig. 8 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Genesio-Tesi chaotic system

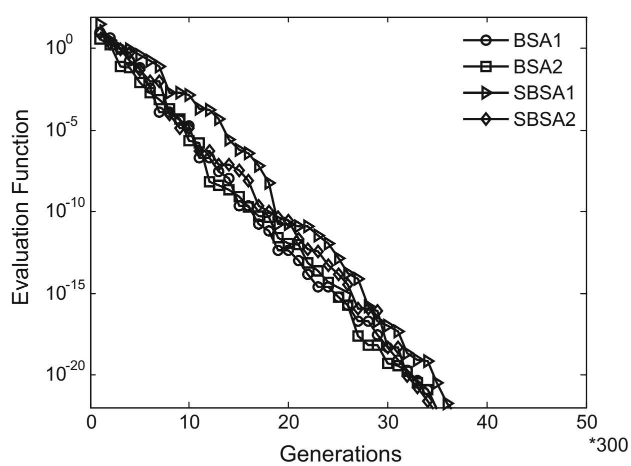

Fig. 9 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Financial chaotic system

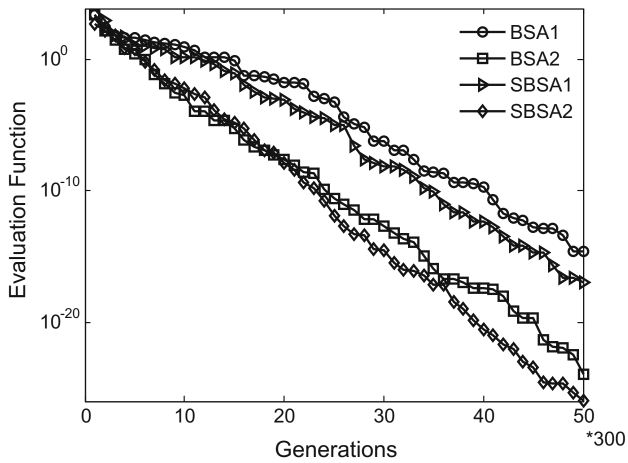

Fig. 10 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Lu chaotic system

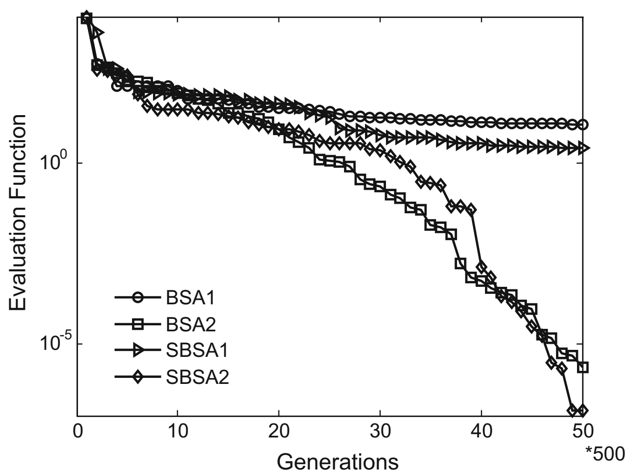

Fig. 11 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Chuas chaotic system

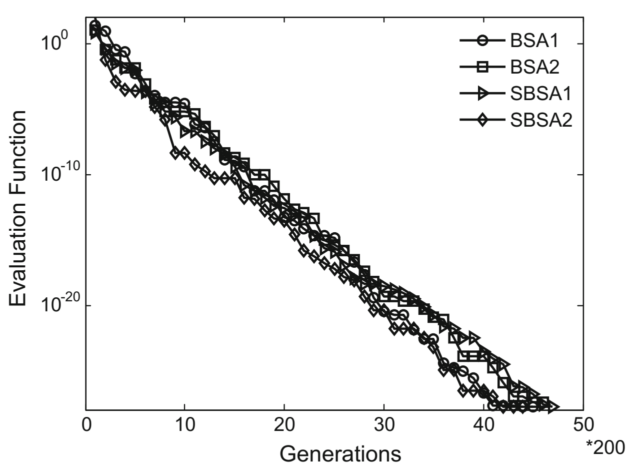

Fig. 12 A comparison among convergence speed of BSA1, BSA2, SBSA1 and SBSA2 for a typical simulation by Henon chaotic system

Table 19 Comparison among results of BSA2 and SBSA2 and results of Tang et al. (2012) on Volta chaotic system (*Note* Zero value on this example means a number smaller than 1.0e−34)

Parameters		θ_1	θ_2	θ_3
True values		36	3	20
Algorithms				
SBSA2	Best	19	11	0.73
	Mean	19	11	0.73
	Best-MSE	0		
	Mean-MSE	0		
BSA2	Best	19	11	0.73
	Mean	19	11	0.73
	Best-MSE	0		
	Mean-MSE	0		
DE	Best	19	11	0.73
	Mean	19	11	0.73
	Best-MSE	0.0000		
	Mean-MSE	3.57e−30		
GA	Best	19	11	0.73
	Mean	19.0074	11.0020	0.7278
	Best-MSE	1.45e−11		
	Mean-MSE	0.0011		

In an overall view, the obtained results of comparison study against results reported in Ho et al. (2010), Tang et al. (2012) and Hu et al. (2015) in Tables 15, 16, 17, 18, 19, 20 and 21 show a considerably better performance of our proposed algorithms. These results confirm efficiency and robustness of two improved versions of BSA than the pure BSA. Also we can conclude that the proposed algorithms in this research can obtain more accurate solutions than the results of other algorithm reported in compared literature.

6 Conclusions

A mathematical model has a vital role in controlling and synchronization of chaotic systems. But generally in real-world problems, parameters are mixed with mismatches and distortions. So presentation of scientific approaches to overcome this drawback is a popular topic. This research proposed two effective ideas to improve performance of BSA. Firstly to provide a faster convergence and a deep search, a new operator to generate initial trial population was proposed. Then a group search ability was provided for the BSA by proposing a SBSA. In the SBSA, a population is divided into several groups and each of groups tries to evolve itself using a BSA evolutionary process. Groping population into several sets

Table 20 Comparison among results of BSA2 and SBSA2 and results of Hu et al. (2015) on Economic chaotic system when orders are unknown (*Note* Zero value on this example means a number smaller than 1.0e−32)

Parameters		α_1	α_2	c
True values		0.9	0.85	1
Algorithms				
SBSA2	Best	0.9	0.85	1
	Mean	0.9	0.85	1
	Worst	0.9	0.85	1
	Best-SSE	2.3419e−19		
	Mean-SSE	5.8331e−16		
	Worst-SSE	5.5262e−15		
BSA2	Best	0.9	0.85	1
	Mean	0.9	0.85	1
	Worst	0.9	0.85	1
	Best-SSE	1.1147e−26		
	Mean-SSE	1.8442e−19		
	Worst-SSE	1.4733e−18		
GABC	Best	0.9	0.85	1
	Mean	0.9	0.85	1
	Worst	0.9	0.85	1
	Best-SSE	1.34E−05		
	Mean-SSE	4.93E−05		
	Worst-SSE	1.06E−04		
EABC	Best	0.9	0.85	1
	Mean	0.9	0.85	1
	Worst	0.9	0.85	1
	Best-SSE	1.58E−06		
	Mean-SSE	1.06E−05		
	Worst-SSE	2.60E−05		
HABC	Best	0.9	0.85	1
	Mean	0.9	0.85	1
	Worst	0.9	0.85	1
	Best-SSE	1.19e−20		
	Mean-SSE	3.22e−09		
	Worst-SSE	4.83e−08		

Table 21 Comparison among results of BSA2 and SBSA2 and results of Hu et al. (2015) on Rossler chaotic system when orders are unknown (*Note* Zero value on this example means a number smaller than 1.0e−32)

Parameters		α_1	α_2	c
True values		0.9	0.85	0.95
Algorithms				
SBSA2	Best	0.9	0.85	0.95
	Mean	0.9	0.85	0.95
	Worst	0.9	0.85	0.95
	Best-SSE	0		
	Mean-SSE	5.3311e−28		
	Worst-SSE	2.5454e−27		
BSA2	Best	0.9	0.85	0.95
	Mean	0.9	0.85	0.95
	Worst	0.9	0.85	0.95
	Best-SSE	0		
	Mean-SSE	2.177e−30		
	Worst-SSE	2.177e−29		
GA	Best	0.9	0.85	0.95
	Mean	0.9	0.85	0.95
	Worst	0.9	0.85	0.95
	Best-SSE		1.44E−02	
	Mean-SSE		2.12E−02	
	Worst-SSE		4.28E−02	
DE	Best	0.9	0.85	0.95
	Mean	0.9	0.85	0.95
	Worst	0.9	0.85	0.95
	Best-SSE		1.20E−08	
	Mean-SSE		2.34E−08	
	Worst-SSE		3.83E−08	
PSO	Best	0.9	0.85	0.95
	Mean	0.9	0.85	0.95
	Worst	0.9	0.85	0.95
	Best-SSE	8.28E−05		
	Mean-SSE	5.57E−04		
	Worst-SSE	2.79E−03		
HABC	Best	0.9	0.85	0.95
	Mean	0.9	0.85	0.95
	Worst	0.9	0.85	0.95
	Best-SSE	3.00e−13		
	Mean-SSE	6.53e−13		
	Worst-SSE	1.11e−12		

provided a better exploration of search space and an independent local search of each groups increased exploitation ability of BSA. Also new proposed operator to generate initial trial population by providing a deep search increased satisfactorily solution quality. To measure efficiency of proposed algorithm with respect to their original versions, they were applied for parameter estimation of 10 typical chaotic systems. The obtained results and nonparametric analysis of them demonstrated that both of the proposed ideas were very effective and robust so that the BSA by aforementioned ideas produced similar and promising results over repeated runs. The BSA with group search ability and new proposed operator called SBSA2 due to a better exploration and a deep exploitation had the best performance among all algorithms. Also the second-best results were related to a BSA with only new proposed operator called BSA2 due to providing a deep search of found promising areas. In the final part of comparison study, a comparison of the proposed algorithms in this study with respect to other algorithms reported in the literature confirmed a considerably better performance of our proposed algorithms.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Humans were not involved in this submission.

References

Ahandani MA, Shirjoposh NP, Banimahd R (2010) Three modified versions of differential evolution algorithm for continuous optimization. Soft Comput 15:803–830

Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10:646–657

Chen D, Chen J, Jiang H, Zou F, Liu T (2014) An improved PSO algorithm based on particle exploration for function optimization and the modeling of chaotic systems. Soft Comput 19:3071–3081

Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219:8121–8144

Duan QY, Gupta VK, Sorooshian S (1993) Shuffled complex evolution approach for effective and efficient global minimization. J Optim Theory Appl 76:501–521

Eusuff MM, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Res Plan Manag 29:210–225

Gao F, Fei FX, Xu Q, Deng YF, Qi YB, Balasingham I (2012) A novel artificial bee colony algorithm with space contraction for unknown parameters identification and time-delays of chaotic systems. Appl Math Comput 219:552–568

Gao F, Lee XJ, Fei FX, Tong HQ, Qi YB, Deng YF, Zhao HL (2013) Parameter identification for Van Der Pol-Duffing oscillator by a novel artificial bee colony algorithm with differential evolution operators. Appl Math Comput 222:132–144

Gao F, Lee XJ, Fei FX, Tong HQ, Deng YF, Zhao HL (2014) Identification time-delayed fractional order chaos with functional extrema model via differential evolution. Expert Syst Appl 41:1601–1608

Ho WH, Chou JH, Guo CY (2010) Parameter identification of chaotic systems using improved differential evolution algorithm. Nonlinear Dyn 61:29–41

Hu W, Yu Y, Zhang S (2015) A hybrid artificial bee colony algorithm for parameter identification of uncertain fractional-order chaotic systems. Nonlinear Dyn 82:1441–1456

Jiang Q, Wang L, Hei X (2015) Parameter identification of chaotic systems using artificial raindrop algorithm. J Comput Sci 8:20–31

Konnur R (2003) Synchronization-based approach for estimating all model parameters of chaotic systems. Phys Rev E 67:027204

Li N, Pan W, Yan L, Luo B, Xu M, Jiang N, Tang Y (2011) On joint identification of the feedback parameters for hyperchaotic systems: an optimization-based approach. Chaos Soliton Fract 44:198–207

Lin J (2015) Oppositional backtracking search optimization algorithm for parameter identification of hyperchaotic systems. Nonlinear Dyn 80:209–219

Lin J (2014) Parameter estimation for time-delay chaotic systems by hybrid biogeography-based optimization. Nonlinear Dyn 77:983–992

Park JH, Kwon OM (2005) A novel criterion for delayed feedback control of time-delay chaotic systems. Chaos Soliton Fract 23:495–501

Sun F, Peng H, Luo Q, Li L, Yang Y (2009) Parameter identification and projective synchronization between different chaotic systems. Chaos Interdiscip J Nonlinear Sci 19:023109

Tang Y, Zhang X, Hua C, Li L, Yang Y (2012) Parameter identification of commensurate fractional-order chaotic system via differential evolution. Phys Lett A 376:457–464

Wang L, Xu Y (2011) An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems. Expert Syst Appl 38:15103–15109

Yu YG, Li HX, Yu JZ (2009) Generalized synchronization of different dimensional chaotic systems based on parameter identification. Mod Phys Lett B 23:2593–2606

Yuan LG, Yang QG (2012) Parameter identification and synchronization of fractional-order chaotic systems. Commun Nonlinear Sci Numer Simul 17:305–316

Zaher A (2008) Parameter identification technique for uncertain chaotic systems using state feedback and steady-state analysis. Phys Rev E 77:036212