Soft Computing
https://doi.org/10.1007/s00500-020-04721-1

METHODOLOGIES AND APPLICATION

A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm

Ali Ghasemi-Marzbali[1]

Abstract

In the recent years, the optimization problems show that they are a big challenge for engineering regarding the fast growth of new nature-inspired optimization algorithms. Therefore, this paper presents a novel nature-inspired meta-heuristic algorithm for optimization which is called as bear smell search algorithm (BSSA) that takes into account the powerful global and local search operators. The proposed algorithm imitates both dynamic behaviors of bear based on sense of smell mechanism and the way bear moves in the search of food in thousand miles farther. Among all animals, bears have inconceivable sense of smell due to their huge olfactory bulbs that manage the sense of different odors. Since the olfactory bulb is a neural model of the vertebrate forebrain, it can make a strong exploration and exploitation for optimization. According to the odors value, bear moves the next location. Therefore, this paper mathematically models these structures. To demonstrate and evaluate the BSSA ability, numerous types of benchmark functions and four engineering problems are employed to compare the obtained results of BSSA with other available optimization methods with several analyzed indices such as pair-wise test, Wilcoxon rank and statistical analysis. The numerical results revealed that proposed BSSA presents competitive and greater results compared to other optimization algorithms.

Keywords Nature-inspired algorithm · Bear smell search algorithm · Benchmark functions · Meta-heuristic algorithm · Bear's sense of smell

1 Introduction

Generally, complex problems like economics, engineering, etc. have some important control variables so that their proper value directly affects on system output. Therefore, the exact setting and getting a proper solution require an expert designer while it may be boring, time-consuming and progression details. Therefore, it is reasonable to convert them as an optimization model that they can solve by the intelligence algorithms (Liu et al. 2018). According to the aforementioned reasons, several techniques have been suggested, and based on their structures, they can be classified into two main groups: classical and inspired-nature optimization algorithms (Zhang et al. 2018; Kohli and Arora 2018; Patel and Savsani 2015; Hashim et al. 2019).

The classical methods used a linear structure with a smooth form to find the best solution. Gradient-based optimization (GO), response surface (RS) methods and Monte Carlo simulations (MCS) (Zhou and Huang 2018), simplex methods (Cerdà et al. 2016), random search (Yang et al. 2018), etc. can be considered in this group.

Since the optimization problems have many local optimal solutions and the nonlinear pattern, they cannot successfully find the best or global solution and may be trapped in a local solution. In another side, the engineering problems are usually limited by many constraints; hereby, they can make difficult conditions for this group to find the accurate solution. According to these points, the classical methods cannot guarantee the final solution whether that is the best answer or not. Therefore, these shortcomings create an incentive for researchers to provide new methods with greater accuracy.

Communicated by V. Loia.

✉ Ali Ghasemi-Marzbali
ali.ghasemi@ustmb.ac.ir

[1] Department of Computer and Electrical Engineering, Mazandaran University of Science and Technology, Babol, Iran

Published online: 01 February 2020

Fortunately, nature offers great opportunities for researchers to provide new artificial computational algorithms to solve the complex problems which mimic the animal's behaviors, plants and others (Zhou et al. 2018).

Therefore, the second group consists of population-based or nature-inspired heuristic algorithms. Having a fair comparison between them, it can conclude that the nature-inspired heuristic algorithms are more capable dealing the complex optimization problems considering the uncertainty, conflicting objectives, partially true (Mortazavi et al. 2018). There are many methods to consider in this group, shorting speaking, some of them are presented below. Genetic algorithm (GA) is one of the random-based algorithms. GA has been very successful in solving linear and some similar problems. GA uses crossover, selection, mutation and reproduction operators to generate next populations (Sirohi et al. 2018).

Particle swarm optimization (PSO) is a global minimization method; it can deal with optimization problems which the solution is a surface or a point in the *n*-dimensional space. Particles based on their velocity coefficients are moved to the next position. It has shown its high ability in solving the continuous optimization problems (dos Santos Júnior and do MonteLima 2018). According to Newton's law of gravity in nature, the gravitational search algorithm (GSA) is presented (Rashedi et al. 2018). In this algorithm, search agents are a set of objects that can be considered as planets of a system. The optimal area, like a black hole, absorbs all planets toward itself. The obtained information of each object is stored in the form of gravitational masses and inertia (Rashedi et al. 2018).

The artificial bee colony (ABC) is another population-based algorithm that inspired bee's honey bee eating behavior. It performs a special type of operator in the local search that combines the random searches for combination or functional optimization (Bansal et al. 2018). According to bee's behavior, honey bee mating optimization (HBMO) is proposed for optimization inspiring the mating process in real bees (Ghasemi et al. 2016). This algorithm mimics the mating process between male bees and queen. The behavior of the bee is a reciprocal relationship between genetics, the physiological and ecological environments and the social condition of the hive or a combination of all those (Ghasemi 2013).

The harmony search algorithm (HSA) is another successful exploration algorithm that is inspired by the simultaneous playback process of the music orchestra during the optimal search process (Valipour and Ghasemi 2017). Gray wolf optimizer (GWO) is a metacognitive algorithm that imitated the hierarchical structure and social behavior of wolves during hunting. It is based on population and has a simple process setting and can be restored to large-scale problems (Long et al. 2018). Cuckoo search algorithm is another optimization algorithm which widely used in different fields of optimization. It is inspired by the lifestyle of a bird named cuckoo (Mareli and Twala 2018). The shuffled frog leaping algorithm (SFLA) is one of the learning optimization algorithms; it imitated the social behavior of the frogs and is categorized among behavioral algorithms or memetic algorithms (Li et al. 2018). Shark smell optimization (SSO) as an optimization method of the recent years is formulated based on the skill of the shark finding the odor source (Abedinia et al. 2016).

Although natural-based algorithms have shown successful performance, beside of them, some algorithms such as league championship algorithm (LCA) are proposed based on non-natural motives (Alimoradi and Kashan 2018); it looks that they are weaker than nature-inspired algorithms. LCA is a stochastic population-based method for continuous global optimization problems which attempts to imitate a championship setting in which artificial teams play in an artificial league for quite a lot of iterations. According to bear manner inspiration, polar bear optimization algorithm (PBO) is proposed in Połap and Woźniak (2017). PBO imitates the polar bears hunt to stay alive in harsh arctic situations and models the way polar bears move in the search of food.

As a comprehensive review of the recent literatures from the well-known database such as IEEE, Elsevier and Springer, it can be easily obvious that there are more algorithms to introduce. In facts, an optimization algorithm consists of several concepts, ideas and operators. Therefore, none of the researchers can claim that their presented algorithms do not need more modifications (Ghasemi et al. 2016).

On the other hand, today's real optimization problem becomes complicated than before because there are not enough data to make a complete mathematic model of a problem in high-dimensional spaces and under the noisy environments (Shayeghi et al. 2017). It is a reasonable cause to prove that there is still a need for another powerful optimization algorithm. Furthermore, based on the "no free lunch" theorem, the presented optimization algorithms cannot apply to all types of optimization (Wolpert and Macready 1997). To the best review knowledge and shorting speaking, some of the presented optimization algorithms can be summarized by Fig. 1 (Liu et al. 2018; Zhang et al. 2018; Kohli and Arora 2018; Patel and Savsani 2015; Hashim et al. 2019; Zhou and Huang 2018; Cerdà et al. 2016; Yang et al. 2018; Zhou et al. 2018; Mortazavi et al. 2018; Sirohi et al. 2018; dos Santos Júnior and do MonteLima 2018; Rashedi et al. 2018; Bansal et al. 2018; Ghasemi et al. 2016; Ghasemi 2013; Valipour and Ghasemi 2017; Long et al. 2018; Mareli and Twala 2018; Li et al. 2018; Abedinia et al. 2016; Alimoradi and Kashan 2018; Połap and Woźniak 2017; Ghasemi et al. 2016).

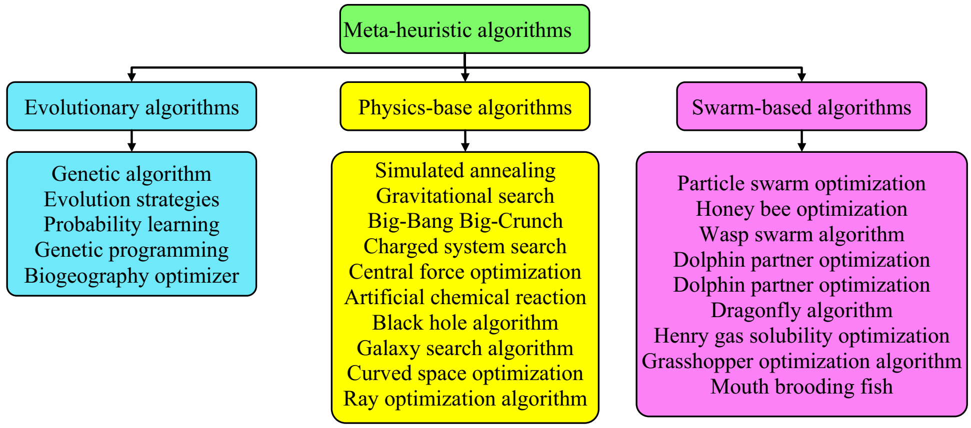

Fig. 1 Meta-heuristic optimization techniques developed in the literature

Some of the optimization algorithms are effective for the special types of optimization while they may be ineffective in other types. Therefore, the "no free lunch" theorem opens this domain for researchers to improve and propose optimization algorithms (Rashedi et al. 2018). These points make a good incentive for researchers to present more and more powerful algorithms.

According to the comprehensive study of animal's behavior and their senses in nature (Liu et al. 2018; Zhang et al. 2018; Kohli and Arora 2018; Patel and Savsani 2015; Hashim et al. 2019), it can be concluded that the bear's sense of smell is stronger than other available intrinsic senses in nature. Bear's olfactory bulb is several times larger than other animals, while its main task is sending of smell information from the nose to the brain. In this method, the bear's sense of smell makes a good way to find food in thousand miles farther (or global solution in optimization). Since bears cannot see food at far distances, the mathematical model based on the sense of smell suggests a powerful way to find the target. Also, this sense is integrated with the neural network; therefore, algorithm learning is increased. According to the aforementioned descriptions, the main contributions and novelties of this paper can be summarized as follows:

(i) A novel nature-inspired bear smell search algorithm is proposed. The olfactory and neuron behavior of smell mechanism is studied thoroughly and modeled mathematically. Also, it proposed the time-varying control parameters to refuse the many control setting and obtain a simple and efficient model.

(ii) The proposed algorithm is evaluated on many benchmark functions to show its performance on different types of local and global search spaces.

(iii) Rigorous comparative study is done with the recent existing nature-inspired optimization algorithms using statistical analysis, convergence rate analysis, ANOVA, etc.

(iv) Robustness and effectiveness of BSSA as well as other optimizers are investigated for four engineering problems.

The next sections of this paper are partitioned as follows. Section 2 presents all mathematic details for the proposed algorithm. Section 3 presents the statistical results and the related discussions on several benchmark functions and engineering problems. Finally, conclusions are given in Sect. 4.

2 Bear smell search algorithm (BSSA)

For reader's convenience, the BSSA is organized as the following subsections in more details.

2.1 Bear's Smell of Sense Mechanism

To predict the quality of an odor from a set of odorant components taking into account, their interaction is a difficult problem, while it is easily done by bear's sense of smell mechanism. In facts, the olfactory bulb is main part of this process (Jordan et al. 2018). Since bears have the larger olfactory bulb compared to other organisms, it makes the great smell of sense. The olfactory bulb receives odor and transmits their information by olfactory tract to the brain. Also, it has the simplest structure between all of the senses causing the simplicity of its cortical model, its signal-processing role and its initial condition phylogenetically. Note that this is a positive point in the proposed algorithm. A complete model of the olfactory system with its components, i.e., glomerular and mitral cell layers is shown in Fig. 2; in fact, it is the sensory organization employed for smelling and known as "olfaction." Shorting speaking, more information about this sense can be given in Li (1990).

As shown in Fig. 2, this model should calculate the glomerular activity relation between odor components as

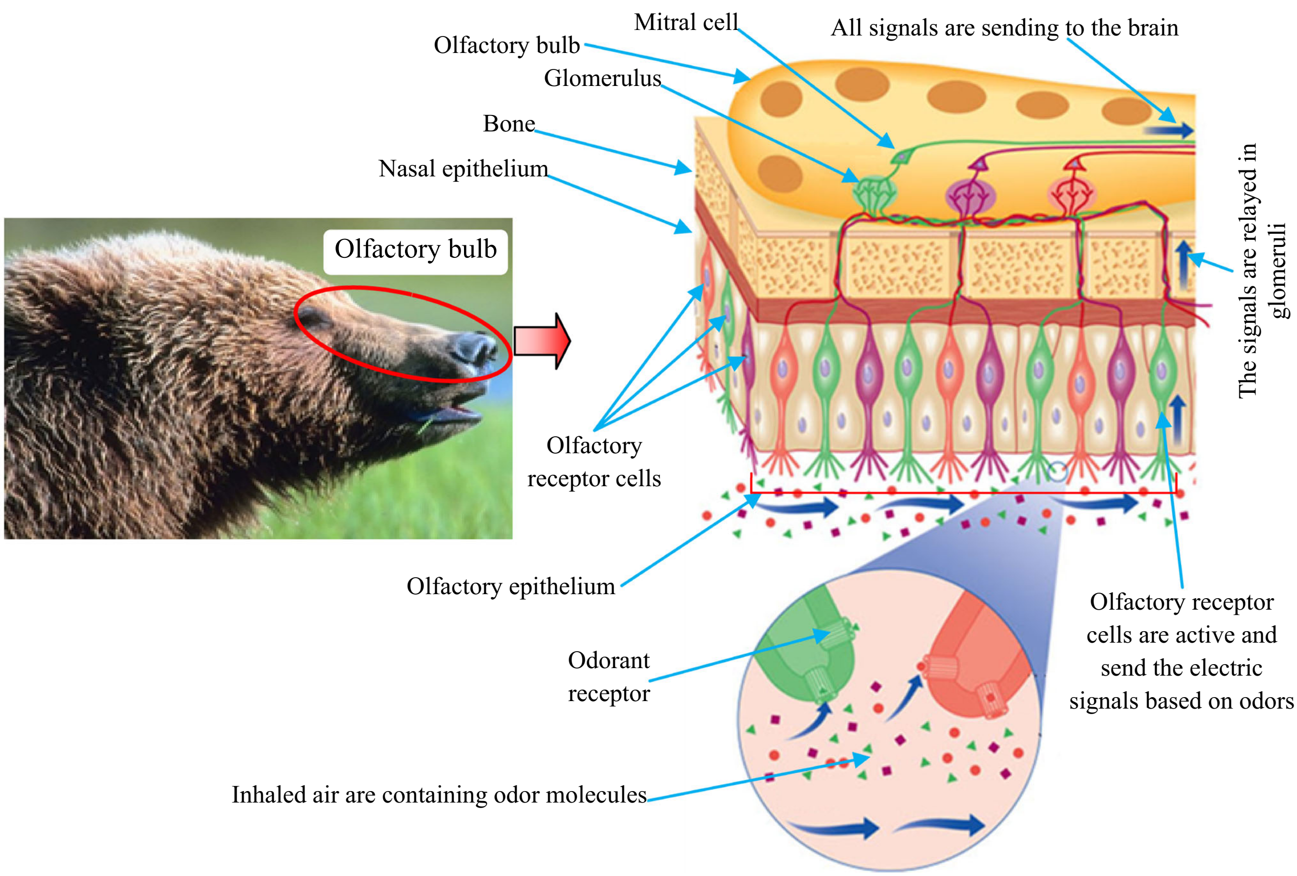

Fig. 2 Overall view of olfactory system, olfactory receptor (ORs) (odorant receptor or olfactory cells) is focused sensory neurons to detect odor components in the lining of the nose; they are bipolar neural cells and derived from the central nervous system and about 100 million of these cells are located in the olfactory epithelium in the vicinity of the retaining cells

inputs. In fact, it makes similarity indices between received odors. Bears can move to the next position based on their similarity values. In order to make better understand for readers and to avoid the unnecessary details of this discussion while those are not within the scope of this article, the main parts of this sense can be summarized as the following parts:

Glomerular part: This layer consists of several glomerular units, tasking measures the glomerular activity pattern on the odorant components. The output of this layer is a matrix of odor components, while it is an input data for the mitral and granular layers.

Mitral and Granular part: The positions of mitral cells are under the glomerular layer. Each one of the mitral and granular cells transmits an unbranched dendrite in a single glomerulus, which fills a large amount of the glomerulus it lies within. Also, the granule cells are below of mitral cell layer in a thick layer. The mitral cell bodies lie in a layer up the granule cell bodies. Each one of the granule cell has an upper dendritic tree that ramifies and finishes in the external plexiform layer.

Dissimilarity assessment part: This part can be known as the decision engine and modeled based on neural network. In other words, brain functions evaluate the odor quality whose neural activity sent through mitral to the olfactory cortex (Yamazaki et al. 1999). Therefore, the dissimilar assessment values can be considered as an index for bears to select their proposed odor or their popular food's odor among the received odors.

2.2 Mathematics formulation of BSSA

This subsection presents the mathematical formulation of BSSA. At first, suppose that bear's nose absorbed different odors so that each one shows a position for moving because everything has a special smell in the environment. Note that many of them are named as the local solution. The particular odor of desired food is the final solution and considered as the global solution. Let $O_i = \begin{bmatrix} oc_i^1 & oc_i^2 & \cdots & oc_i^j & \cdots & oc_i^k \end{bmatrix}$ be the ith received odor with k components or molecules. Since bear receives n odors in breathing time, the initial solution is a matrix, $\mathrm{OM} = [O_i]_{n\times k} = [oc_i^j]_{n\times k}$. Now, according to the

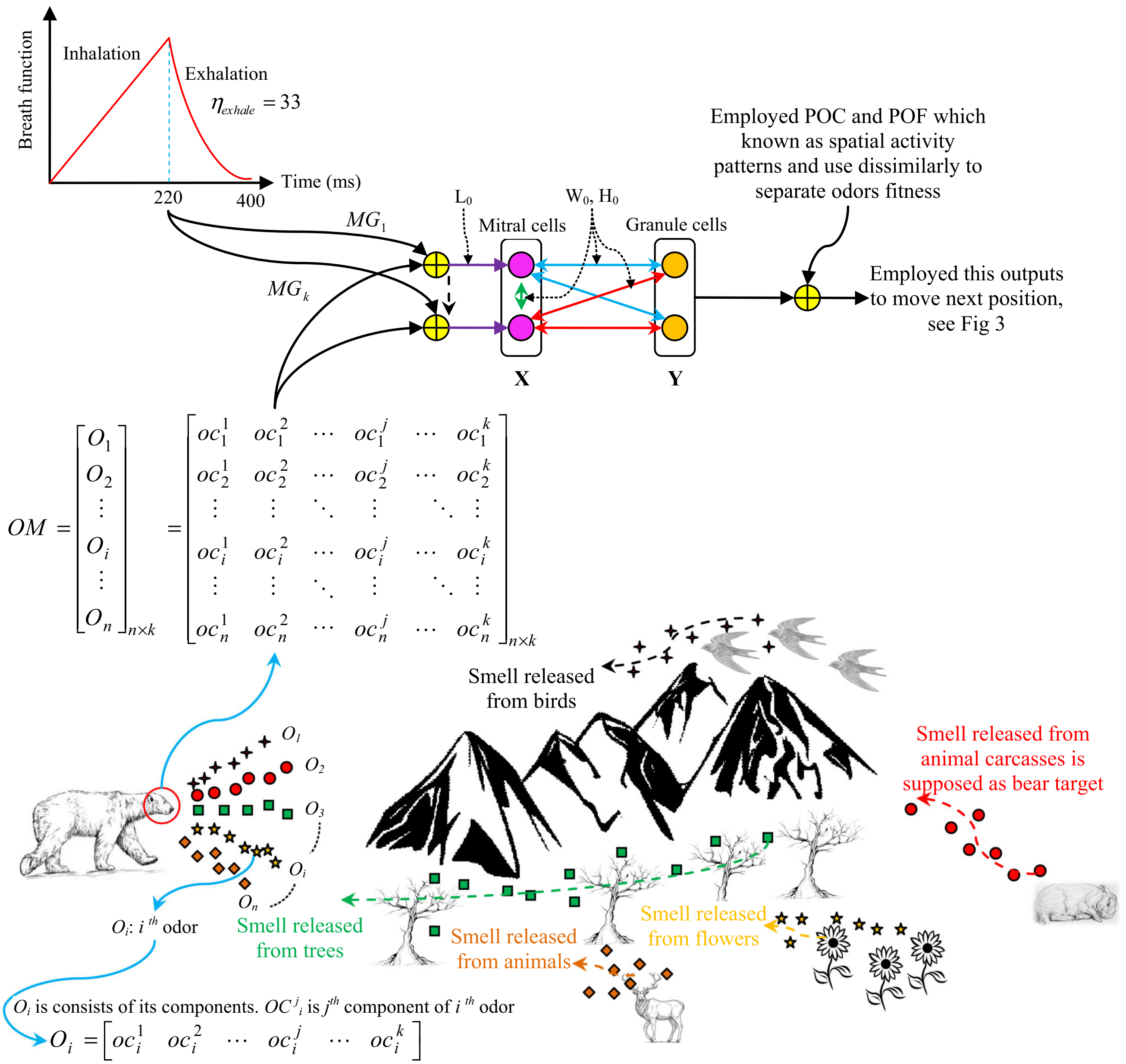

Fig. 3 Graphical view of the olfactory system, since the direction of odors toward bear is important; therefore, only these types of odors are drawn but in real-world these odors released in all directions

glomerular layer process and breathing function in a sniff cycle, let DS_i^j denote the jth odor component in the ith odor. Based on mathematic formulations (Li and Hopfield 1989), one gets:

$$\mathrm{DS}_i^j = \begin{cases} \mathrm{MG}_i(t - t_{\mathrm{inhale}}) + DS_i^{t_{\mathrm{inhale}}}, & t_{\mathrm{inhale}} \le t \le t_{\mathrm{exhale}} \\ \mathrm{DS}_i^{t_{\mathrm{exhale}}} \exp\left(\dfrac{t_{\mathrm{exhale}} - t}{\eta_{\mathrm{exhale}}}\right), & t_{\mathrm{exhale}} \le \mathrm{t} \end{cases} \tag{1}$$

where t_{exhale}, t_{inhale} and η_{exhale} are exhalation time, inhalation time and a constant value for exhalation time, respectively. In the optimization process, the overall time of a cycle of breathing is the same as k or length of ith odor and according to exhalation and inhalation times; odor components are divided into two groups. The allocated time to these parameters is shown in Fig. 3. $\mathrm{MG} = \{\mathrm{MG}_1, \mathrm{MG}_2, \ldots, \mathrm{MG}_i, \ldots, \mathrm{MG}_n\}$ contains the receptor sensitivities and the odor absorption and identity, and they have an input (integrated their information) for the ith mitral. $\mathrm{DS}_i^j = 0$ shows that there is not odor up the olfactory epithelium before next inhalation. The nonnegative set MG can be calculated by (Grossman et al. 2008):

$$\mathrm{MG}_i(O_i) = \frac{1}{k}\sum_{j=1}^{k} f(oc_i^j), f(oc_i^j) = \begin{cases} 1, & T_1 \le oc_i^j \\ 0, & T_1 > oc_i^j \end{cases} \quad (2)$$

where k denotes the odor length in ith odor and T_1 is a threshold variable and sets based on average value of odors information. Now, these data are sent to the mitral and granular layers, in order to achieve this goal, employed the Li–Hopfield and the Erdi formulations which imitate the neural dynamics arising from granular and mitral layers by (Li and Hopfield 1989):

$$\begin{aligned} \dot{X} &= -H_0\Psi_y(Y) - \alpha_x X + \sum L_0\Psi_x(X) + DS \\ \dot{Y} &= W_0\Psi_x(X) - \alpha_y Y + DS_c \end{aligned} \quad (3)$$

where $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$ are mitral and granule cells activities, respectively. $\mathrm{DS} = \{ds_1, ds_2, \ldots, ds_n\}$ and $\mathrm{DS}_c = \{ds_{c1}, ds_{c2}, \ldots, ds_{cn}\}$ are the external input to the mitral and the central input to the granule cells, respectively.

Also, $\Psi_x(X) = \{f_x(x_1), f_x(x_2), \ldots, f_x(x_n)\}$ and $\Psi_y(Y) = \{f_y(y_1), f_y(y_2), \ldots, f_y(y_n)\}$ denote the outputs of the mitral and the granule cells, respectively. α_x and α_y are time constants of mitral and granule cells and their values are 0.14. f_x and f_y simulate the cell output functions for the mitral and granule cells; one gets (Li 1990):

$$f_x(x) = \begin{cases} 0.14 + 0.14\tanh\left(\frac{x-\varphi}{0.14}\right), & x < \varphi \\ 0.14 + 1.4\tanh\left(\frac{x-\varphi}{1.4}\right), & x \ge \varphi \end{cases} \quad (4)$$

$$f_y(y) = \begin{cases} 0.29 + 0.29\tanh\left(\frac{x-\phi}{0.29}\right) & x < \phi \\ 0.29 + 2.9\tanh\left(\frac{x-\phi}{2.9}\right) & x \ge \phi \end{cases} \quad (5)$$

where φ is a threshold value. H_0, W_0 and L_0 are the synaptic-strength connection matrixes which denote the relationship between granular and mitral cells and the relationship between mitral cells and calculate by following formulations (Li 1990):

$${H_0}_i^j = \begin{cases} \frac{\mathrm{rand}()}{T_h}, & 0 < d_i^j < T_h \\ 0, & T_h < d_i^j \end{cases} \quad (6)$$

$${W_0}_i^j = \begin{cases} \frac{\mathrm{rand}()}{T_w}, & 0 < d_i^j < T_w \\ 0, & T_w < d_i^j \end{cases} \quad (7)$$

$${L_0}_i^j = \begin{cases} \frac{\mathrm{rand}()}{T_l}, & 0 < d_i^j < T_l \\ 0, & T_l < d_i^j. \end{cases} \quad (8)$$

T_h, T_w and T_l are the connection constants. rand() is a random value. d_i^j presents the distance between ith and gth odors caused by their information, while gth odor is the desired smell for bear. In other words, this distance is defined between each odor (local solution) and intended odor (global solution). It shows that the guided mechanism based on the global solution is employed during the optimization process to improve the exploitation. According to aforesaid descriptions, when brain obtained all information from neural activity, the separating process begins based on dissimilarity assessment. This process is simulated based on the Pearson correlation. Therefore, this point helps bear to select the best way for the next position. Let the probability odor components (POC), probability odor fitness (POF) and odor fitness (OF) defined by:

$$\mathrm{POC}_i = \frac{O_i}{\max(O_i)} \quad (9)$$

$$\mathrm{POF}_i = \frac{\mathrm{OF}_i}{\max(\mathrm{OF}_i)}. \quad (10)$$

The dissimilarity between two odors can be calculated by expected odor fitness (EOF) and distance odor components (DOC) formulations as follows:

$$\mathrm{DOC}_i = 1 - \frac{\sum_{i=1}^{k}(\mathrm{POC}_j^1 - \mathrm{POC}_j^2)}{\sqrt{\sum_{j=1}^{k}(\mathrm{POC}_j^1 - \mathrm{POC}_j^2)^2}} \quad (11)$$

$$\mathrm{EOF}_i = |\mathrm{POF}_i - \mathrm{POF}^g| \quad (12)$$

where g refers to the global solution. The above equations dictate the possible way to move. In fact, these indices explain the relation between odors which they are arrived at the desired position. This concept can be illustrated by the mesh mechanism as shown in Fig. 4. It is clearly illustrated that the outputs of the brain decide a suitable way for the next position. In each area of the mesh grid, the distance between of all odors is calculated based on two thresholds; ζ_1, ζ_2; thus, the next odors can be calculated by:

$$\begin{aligned} &O_{k+1} = \\ &\begin{cases} C_{1,i} \times O_k - \mathrm{rand} \times C_{2,i} \times (O_k - O_{\mathrm{best}}), & \mathrm{DOC}_i \le \zeta_1 \text{ and } \mathrm{EOF}_i \le \zeta_2 \\ C_{3,i} \times O_k - \mathrm{rand} \times C_{4,i} \times (O_k - O_{\mathrm{best}}), & \text{otherwise} \end{cases} \\ &C_{1,i} = -\mathrm{EOF}_i\frac{2-\mathrm{DOC}_i}{\zeta_1}, C_{2,i} = -\mathrm{EOF}_i\frac{2-\mathrm{DOC}_i}{\zeta_2} \\ &C_{3,i} = \mathrm{EOF}_i\frac{2-\mathrm{DOC}_i}{\zeta_1}, C_{4,i} = \mathrm{EOF}_i\frac{2-\mathrm{DOC}_i}{\zeta_2}. \end{aligned} \quad (13)$$

2.3 Framework of BSSA

In order to summarize the previous formulations and introduce the proposed software, Fig. 5 describes the standard bear smell search algorithm framework.

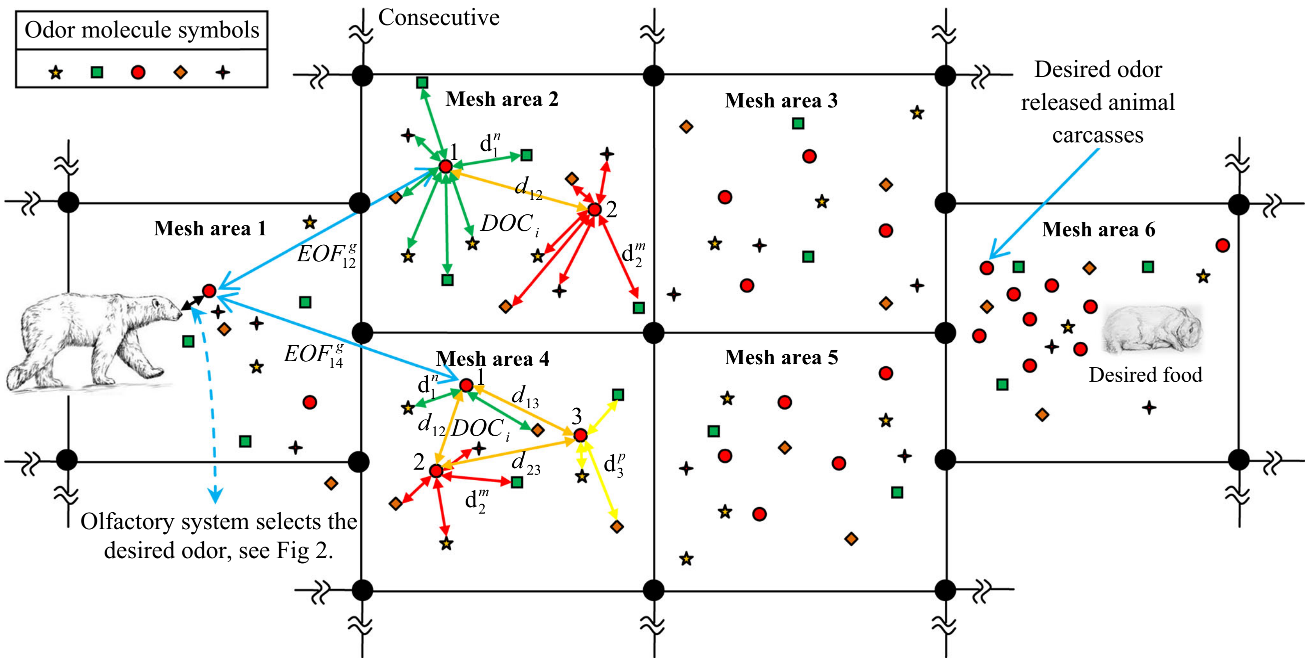

Fig. 4 The overall concept of the mesh mechanism to move next position

3 Experimental study

3.1 Mathematical benchmark problems

This section presents various formulations of mathematical benchmark problems, while most of them are well known for researchers as CEC benchmarks (Hashim et al. 2019; Liang et al. 2013). This paper collects many benchmarks based on similarities in their significant physical properties and forms to evaluate the proposed algorithm under various conditions, e.g., many local solutions, flat plan, etc. Furthermore, several engineering problems are employed to show the outperformance of BSSA compared to other available engineering designs in the last literature.

They have performed on PC with Intel (R) Core (TM) Duo CPU @ 2.53 GHz, 4 GB of RAM using MATLAB 2011a. In order to have a regular categorization based on relative similarities, the employed benchmark functions are classified into six main groups. The mathematical formulations with their class are tabulated in Table 1. It should be noted that the number of evaluated functions is a positive point for this article since there is not the same paper with this number of benchmark functions.

To evaluate and compare the performance of BSSA on the aforementioned benchmark functions, consider the following optimization algorithms, which are selected based on their high potential in search of the optimal solution. They are genetic algorithm (GA) (Sirohi et al. 2018), particle swarm optimization (PSO) (dos Santos Júnior and do MonteLima 2018), gravitational search algorithm (GSA) (Rashedi et al. 2018), artificial bee colony (ABC) (Bansal et al. 2018), honey bee mating optimization (HBMO) (Ghasemi et al. 2016), gray wolf optimizer (GWO) (Long et al. 2018), cuckoo search algorithm (COA) (Mareli and Twala 2018), championship algorithm (LCA) (Alimoradi and Kashan 2018), Henry gas solubility optimization (HGSO) (Hashim et al. 2019) and squirrel search algorithm (SSA) (Jain et al. 2019). In order to have a fair comparison between these algorithms, consider the same initial conditions such as the number of population and the number of iteration. The obtained results by the compared optimization algorithms after 30 independent runs over the listed function in Table 1 are reported in Table 2.

As shown in Table 2 and in the coming tables, the bolded numbers indicate the best obtained results by these algorithms in each benchmark. According to static indices, it can be seen that the BSSA can find best or global solutions for all benchmark functions. Also, other successful algorithms with best searching operators are GSA, SSA, HGSO and GWO. Meanwhile, HGSO and SSA as the newest algorithm have the closest competition with the proposed BSSA method in solving benchmark functions. On average, the genetic algorithm has shown the weakest performance. Table 2 describes the merits of BSSA to balance between exploitation and exploration terms.

In f_{16}, HGSO has better mean value − 1.867E+02 compared to BSSA with − 1.823E+02, about 0.2%. In f_{24}, the worst value by SSA is 8.125E−24, while this value for BSSA is 5.736E−19. Also, for f_{31}, PSO, GWO and SSA have better standard deviation compared to BSSA, while

```
Begin
Set the initial parameters:
        Number of population N
        Maximum iteration Iter_max
        Lower and upper bounds; LB and UB
        Number of variables D
Generate the initial random population with dimension OM = N×D in search space
Evaluate fitness for each row of matrix OM
Save best odor as current global solution, O^g
for iter = 1 : Iter_max
        Calculate the maximum output of the glomerular activity by Eq (2); MG
        Obtain DS based on breathing function in Eq (1)
        Calculate the cell output functions for the mitral and granule cells, f_x(x) and f_y(y), Eqs (4)-(5)
        Define H_0, W_0 and L_0 based on Eqs (6)-(8)
        Solve Eq (3)
        Set vectors POC, POF and OF based on Eqs (9)-(10)
        Calculate vectors EOF and DOC by Eqs (11)-(12)
        Set two thresholds ζ_1 and ζ_1
        Calculate the coefficients C_1 to C_4
        for i = 1 : N
                if Satisfy the criterion then
                        Generate new population based on first part of Eq (13)
                else
                        Generate new population based on second part of Eq (13)
                end
        end
end
Draw and print the desired outputs
```

Fig. 5 Pseudocode for BSSA

they were not successful to find the global solution because trapped to a local solution and BSSA finds very better solution compared to them. In f_{32}, ABC obtained better result comparing other algorithms by record 2.511E−02 in minimum position, while this value for BSSA is 2.514E−02; these values are very close, but ABC was weakest in the other indices. In f_{38}, only for standard deviation index, SSA records better value. Similarly, in f_{44}, COA has a better standard deviation. Since these benchmark functions include many local solutions, they are employed used to examine the ability of proposed algorithms to clearly show their local and global searches. To get the quantitative analysis between the employed algorithms, mean absolute error (MAE) index is used for some of test functions. It is an easy, safe and a valid statistical criterion to rank the optimization algorithms based on difference between algorithm results of actual values and can be formulated by:

$$\mathrm{MAE} = \frac{1}{N}\sum\nolimits_{j=1}^{N} \left|Op_j - G_j\right| \tag{14}$$

where Op_j and G_j are average of optimal solutions and global solution, respectively. N is the number of test functions. Table 3 lists the average error rates obtained in the 11 benchmark functions. According to selected benchmark functions of Table 3, the ranking of employed algorithms based on their corresponding MAE's is tabulated in Table 4.

However, BSSA was not absolutely better for all indices in all benchmark functions, but it was very powerful in most of them. Since there are many benchmark functions and Tables 2, 3 and 4 represent complete analyses, it is not reasonable to show the graphical results for all of them. Therefore, Figs. 6, 7, 8, 9, 10, 11 and 12 show the graphical results for convergence and ANOVA tests for some of them. Note that the ANOVA test shows variance deviation analyses; therefore, it creates a good comparison between them. Moreover, these figures do not include the same benchmark functions so as to cover most of them. These figures present a graphical view for reader convenience with little effort or difficulty to compare algorithms performance. These results clearly show that the BSSA has better convergence with least iteration. It can be obvious that BSSA is a fast and reliable algorithm even if it was not successful in a few indices. In contrast, the proposed method can find better and actual solution well and fast, while other methods still have a significance gap to the actual solution. Related to the length of available data, the

Table 1 The mathematical detail of employed benchmark functions

No.	Fun	[L,U]	D	Formulation	Min
Many local minima group					
1	Ackley	[– 32, 32]	30	$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + \exp(1)$	0
2	Bukin N. 6	$x_1 \in [-15, -5]$ $x_2 \in [-3, 3]$	2	$f_2(x) = 100\sqrt{\lvert x_2 - 0.01x_1^2\rvert} + 0.01\lvert x_1 + 10\rvert$	0
3	Cross-in-tray function	[– 10, 10]	2	$f_3(x) = -0.0001\left(\left\lvert\sin(x_1)\sin(x_2)\exp\left(\left\lvert 100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert\right)\right\rvert + 1\right)^{0.1}$	– 2.06261
4	Drop wave	[– 5.12, 5.12]	2	$f_4(x) = \frac{1+\cos\left(12\sqrt{x_1^2+x_2^2}\right)}{0.5\left(x_1^2+x_2^2\right)+2}$	– 1
5	Eggholder	[– 512, 512]	2	$f_5(x) = -(x_2+47)\sin\left(\sqrt{\left\lvert x_2 + \frac{x_1}{2} + 47\right\rvert}\right) - x_1\sin\left(\sqrt{\lvert x_1 - x_2 - 47\rvert}\right)$	– 959.6407
6	Gramacy & Lee (2012)	[0.5, 2.5]	1	$f_6(x) = \frac{\sin(10\pi x)}{2x} + (x-1)^4$	– 0.8690
7	Griewank	[– 600, 600]	30	$f_7(x) = \sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0
8	Holder table	[– 10, 10]	2	$f_8(x) = -\left\lvert\sin(x_1)\cos(x_2)\exp\left(\left\lvert 1 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert\right)\right\rvert$	– 19.2085
9	Langermann	[0, 10]	10	$f_9(x) = -\sum_{i=1}^{m} c_i\exp\left(-\frac{1}{\pi}\sum_{j=1}^{D}(x_j - A_{ij})^2\right)\cos\left(\pi\sum_{j=1}^{D}(x_j - A_{ij})^2\right), m = 5, c = (1,2,5,2,3), A = \begin{pmatrix} 3 & 5 & 2 & 1 & 7 \\ 5 & 2 & 1 & 4 & 9 \end{pmatrix}^T$	– 1.4
10	Levy	[– 10, 10]	5	$f_{10}(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{D-1}(\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_i + 1)] + (\omega_d - 1)^2[1 + \sin^2(2\pi\omega_d)], \omega_i = 1 + \frac{x_i - 1}{4}$	0
11	Levy N. 13	[– 10, 10]	2	$f_{11}(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2[1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + \sin^2(2\pi x_2)]$	0
12	Rastrigin	[– 5.12, 5.12]	30	$f_{12}(x) = 10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$	0
13	Schaffer N. 2	[– 100, 100]	2	$f_{13}(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	0
14	Schaffer N. 4	[– 100, 100]	2	$f_{14}(x) = 0.5 + \frac{\cos^2(\sin(\lvert x_1^2 - x_2^2\rvert)) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	0.292579
15	Schwefel	[– 500, 500]	30	$f_{15}(x) = 418.9329D - \sum_{i=1}^{D} x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	0
16	Shubert	[– 10, 10]	2	$f_{16}(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$	– 186.7309
Bowl-shaped group					
17	Bohachevsky	[– 100, 100]	2	$f_{17}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	0
18	Perm, Beta	[– 10, 10]	10	$f_{18}(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{D}(j + \beta)\left(x_j^i - j^{-i}\right)\right)^2$	0
19	Rotated Hyper-Ellipsoid	[– 65.536, 65.536]	30	$f_{19}(x) = \sum_{i=1}^{D}\sum_{j=1}^{i} x_j^2$	0

Table 1 (continued)

Many local minima group					
No.	Fun	[L,U]	D	Formulation	Min
20	Sphere	[− 5.12, 5.12]	30	$f_{20}(x)=\sum_{i=1}^{D}x_i^2$	0
21	Sum of different powers	[− 100, 100]	30	$f_{21}(x)=\sum_{i=1}^{D}\lvert x_i\rvert^{i+1}$	0
22	Sum squares	[− 10, 10]	30	$f_{22}(x)=\sum_{i=1}^{D}ix_i^2$	0
23	Trid	[− 100, 100]	10	$f_{23}(x)=\sum_{i=1}^{D}(x_i-1)^2-\sum_{i=2}^{D}x_ix_{i-1}$	− 210
Plate-shaped group					
24	Booth	[− 10, 10]	2	$f_{24}(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	0
25	Matyas	[− 10, 10]	2	$f_{25}(x)=0.26(x_1^2+x_2^2)-0.48x_1x_2$	0
26	McCormick	$x_1\in[-1.5, 4]$ $x_2\in[-3, 4]$	2	$f_{26}(x)=\sin(x_1+x_2)+(x_1-x_2)^2-1.5x_1+2.5x_2+1$	− 1.9133
27	Power sum	[0, 4]	4	$f_{27}(x)=\sum_{i=1}^{D}\left[\left(\sum_{j=1}^{D}x_j^i\right)-b_i\right]^2, b=(8, 18, 44, 114)$	0
28	Zakharov	[− 5, 10],	10	$f_{28}(x)=\sum_{i=1}^{D}x_i^2+\left(\sum_{i=1}^{D}0.5ix_i\right)^2+\left(\sum_{i=1}^{D}0.5ix_i\right)^4$	0
Valley-shaped group					
29	Three-hump camel	[− 5, 5]	2	$f_{29}(x)=2x_1^2-1.05x_1^4+\frac{x_1^6}{6}+x_1x_2+x_2^2$	0
30	Six-hump camel	$x_1\in[-3, 3]$ $x_2\in[-2, 2]$	2	$f_{30}(x)=\left(4-2.1x_1^2+\frac{x_1^4}{3}\right)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$	− 1.0316
31	Dixon-price	[− 10, 10]	10	$f_{31}(x)=(x_1-1)^2+\sum_{i=2}^{D}i(2x_i^2-x_{i-1})^2$	0
32	Rosenbrock	[− 5, 10]	10	$f_{32}(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	0
Steep ridges/drops group					
33	De jong N. 5	[− 65.536, 65.536]	2	$f_{33}(x)=\left(0.002+\sum_{i=1}^{25}\frac{1}{i+(x_1-a_{1i})^6+(x_2-a_{2i})^6}\right)^{-1}$, $a=\begin{pmatrix}-32 & -16 & 0 & 16 & 32 & -32 & \cdots & 0 & 16 & 32\\ -32 & -32 & -32 & -32 & -32 & -16 & \cdots & 32 & 32 & 32\end{pmatrix}$	0.9980
34	Easom	[− 100, 100]	2	$f_{34}(x)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	− 1
35	Michalewicz	[0, π]	5	$f_{35}(x)=-\sum_{i=1}^{D}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$	− 4.687658
Noisy group					
36	Beale	[− 4.5, 4.5]	2	$f_{36}(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2^2)^2+(2.625-x_1+x_1x_2^3)^2$	0

Table 1 (continued)

Many local minima group					
No.	Fun	[L,U]	D	Formulation	Min
37	Branin	$x_1 \in [-5, 10]$ $x_2 \in [0, 15]$	2	$f_{37}(x) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)\cos(x_1) + s,$ $a = 1,\ b = 5.1/(4\pi^2),\ c = 5/\pi,\ r = 6,\ s = 10,\ t = 1/(8\pi)$	0.397887
38	Colville	[− 10, 10]	4	$f_{38}(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2$ $+10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$	0
39	Forrester et al. (2008)	[0, 1]	1	$f_{39}(x) = (6x - 2)^2 \sin(12x - 4)$	− 6.0207
40	Goldstein-price	[− 2, 2]	2	$f_{40}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3
41	Hartmann 3-D	[0, 1]	3	$f_{41}(x) = -\sum_{i=1}^{4}\alpha_i \exp\left(-\sum_{j=1}^{3}A_{ij}(x_j - P_{ij})^2\right), \alpha = (1, 1.2, 3, 3.2)^T,$ $A = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, P = 10^{-4}\begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix}$	− 3.86278
42	Hartmann 4-D	[0, 1]	4	$f_{42}(x) = \frac{1}{0.839}\left[1.1 - \sum_{i=1}^{4}\alpha_i \exp\left(-\sum_{j=1}^{4}A_{ij}(x_j - P_{ij})^2\right)\right],$ $A = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 01 & 14 \end{bmatrix},$ $P = 10^{-4}\begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix}, \alpha = (1, 1.2, 3, 3.2)^T$	− 3.13
43	Hartmann 6-D	[0, 1]	6	$f_{43}(x) = -\sum_{i=1}^{4}\alpha_i \exp\left(-\sum_{j=1}^{6}A_{ij}(x_j - P_{ij})^2\right),$ $A = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 01 & 14 \end{bmatrix},$ $P = 10^{-4}\begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix}$ $, \alpha = (1, 1.2, 3, 3.2)^T$	− 3.0425

Table 1 (continued)

Many local minima group					
No.	Fun	[L,U]	D	Formulation	Min
44	Perm D, β	[− 4, 4]	4	$f_{44}(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{D}(j^i+\beta)\left(\left(\frac{x_j}{j}\right)^i-1\right)\right)^2, \beta=0.5$	0
45	Powell	[− 4, 5]	10	$f_{45}(x)=\sum_{i=1}^{D/4}[(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4]$	0
46	Shekel	[0, 10]	4	$f_{46}(x)=-\sum_{i=1}^{m}\left(\sum_{j=1}^{4}(x_j-C_{ji})^2+\beta_i\right)^{-1}, m=10, \beta=0.1(1,2,2,4,4,6,3,7,5,5)^T,$ $C=\begin{bmatrix}4&1&8&6&3&2&5&8&6&7\\4&1&8&6&7&9&3&1&2&3\\4&1&8&6&3&2&5&8&6&7\\4&1&8&6&7&9&3&1&2&3\end{bmatrix}$	− 10.5364
47	Styblinski-Tang	[− 5, 5]	10	$f_{47}(x)=\frac{1}{2}\sum_{i=1}^{D}(x_i^4-16x_i^2+5x_i)$	− 391.66
48	Quartic	[− 1.28, 1.28]	40	$f_{48}(x)=\sum_{i=1}^{n}ix_i^4+random[0,1]$	0
49	Egg Crate	[− 5, 5]	2	$f_{49}(x)=x_1^2+x_2^2+25(\sin^2(x_1)+\sin^2(x_2))$	0
50	Chichinadze	[− 30, 30]	2	$f_{50}(x)=x_1^2+12x_1+11+10\cos(\frac{\pi x_1}{2})+8\sin(\frac{5\pi x_1}{2})-0.2^{0.5}\exp(-0.5(x_2-0.5)^2)$	− 43.3159
51	Sawtoothxy	[− 20, 20]	2	$f_{51}(x)=g(r).h(t), \text{where}, g(r)=\left[\sin(r)-\frac{\sin(2r)}{2}+\frac{\sin(3r)}{3}+\frac{\sin(4r)}{4}+4\right]\left(\frac{r^2}{r+1}\right)$ $h(t)=0.5\cos(2t-0.5)+\cos(t)+2, r=\sqrt{x_1^2+x_2^2}, t=a\tan^2(x_1,x_2)$	0
52	Rump	[− 500, 500]	2	$f_{52}(x)=(333.75-x_1^2)x_2^6+x_1^2(11x_1^2x_2^4-2)+5.5x_2^8+\frac{x_1}{2x_2}$	0
53	Venter	[− 50, 50]	2	$f_{53}(x)=x_1^2-100\cos(x_1^2)-100\cos\left(\frac{x_1^2}{30}\right)+x_2^2-100\cos(x_2^2)-100\cos\left(\frac{x_2^2}{30}\right)$	− 400

D dimension, *[L, U]* lower and upper bands, *Fun* function name, *No* number, *min* minimum value

Table 2 Statistical results obtained by GA, PSO, GSA, ABC, HBMO, GWO, COA, LCA, HGSO and SSA through 30 independent runs on mentioned benchmark functions in Table 1

No.	Algorithms indices	GA	PSO	GSA	ABC	HBMO	GWO
f_1	Best	2.142E+00	7.053E−03	6.983E−06	5.982E−10	2.897E−03	1.065E−10
	Worst	3.525E+00	1.279E−01	3.982E−02	3.043E−06	1.432E+00	4.873E−03
	Mean	3.240E+00	9.858E−02	0.195E−01	0.989E−09	2.761E−02	7.565E−06
	STD	3.058E−01	1.586E−02	1.329E−01	1.093E−02	1.076E−01	3.762E−02
f_2	Best	7.837E+01	5.381E+00	2.098E+00	4.652E+00	3.323E+00	2.546E−03
	Worst	9.651E+02	9.043E+01	8.675E+00	8.341E+00	4.782E+00	4.726E−02
	Mean	8.564E+01	7.542E+01	3.342E+00	6.783E+00	3.895E+00	1.657E−02
	STD	1.243E+01	1.054E+01	1.764E−01	3.762E+00	2.746E+00	1.323E−01
f_3	Best	− 0.323E+00	− 0.645E+00	− 1.972E+00	− 1.234E+00	− 1.892E+00	− 1.997E+00
	Worst	1.248E+00	1.121E+00	− 1.146E+00	0.653E+00	2.619E−01	− 1.143E+00
	Mean	1.032E+00	0.342E+00	− 1.345E+00	− 8.837E−01	− 1.125E+00	− 1.657E+00
	STD	8.637E+01	9.748E+00	1.097E−01	2.019E−01	3.524E−01	5.847E−02
f_4	Best	− 1.425E−01	− 4.837E−01	**− 1.000E+00**	− 8.948E−01	− 9.924E−01	**− 1.000E+00**
	Worst	3.243E+00	− 1.093E−01	− 9.849E−01	− 2.192E−01	− 5.423E−01	− 7.837E−01
	Mean	1.907E−01	− 1.623E−01	− 9.992E−01	− 7.765E−01	8.897E−01	− 8.879E−01
	STD	1.783E+01	5.425E−01	− 6.938E−05	1.492E−02	1.029E−02	5.024E−03
f_5	Best	− 7.653E+02	− 9.321E+02	− 9.585E+02	**− 9.596E+02**	− 9.594E+02	**− 9.596E+02**
	Worst	− 6.536E+02	− 9.025E+02	− 9.413E+02	− 9.513E+02	− 9.389E+02	− 9.563E+02
	Mean	− 7.192E+02	− 9.294E+02	− 9.526E+02	− 9.576E+02	− 9.951E+02	− 9.578E+02
	STD	3.201E+00	3.052E−01	1.0451E−02	3.656E−02	1.443E−03	1.493E−04
f_6	Best	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**
	Worst	− 8.689E−01	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**
	Mean	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**
	STD	0.000E−06	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_7	Best	1.208E−01	**0.000E+00**	**0.000E+00**	4.837E−08	8.921E−06	**0.000E+00**
	Worst	2.685E−01	1.029E+00	2.546E−02	2.039E−01	6.209E−01	4.029E−03
	Mean	1.952E−01	1.938E−02	1.645E−05	5.038E−07	5.838E−05	7.983E−05
	STD	3.915E−02	1.523E−03	5.320E−04	9.456E−04	9.487E−04	2.544E−05
f_8	Best	− 11.414E+00	− 13.059E+00	− 17.894E+00	− 18.021E+00	− 17.890E+00	− 19.011E+00
	Worst	− 1.763E+00	− 3.572E+00	− 11.527E+00	− 9.920E+00	− 13.922E+00	− 15.829E+00
	Mean	− 5.827E+00	− 7.529E+00	− 16.820E+00	− 16.132E+00	− 16.153E+00	− 18.930E+00
	STD	2.516E+01	1.926E+01	1.209E+01	2.837E+01	1.213E+01	6.920E+00
f_9	Best	− 1.021E+00	− 1.382E+00	− 1.396E+00	− 1.388E+00	− 1.399E+00	− 1.393E+00
	Worst	− 1.763E−01	− 4.029E−01	− 9.635E−01	− 9.077E−01	− 9.101E−01	− 1.111E+00
	Mean	− 6.389E−01	− 7.625E−01	− 1.212E+00	− 1.354E+00	− 1.298E+00	− 1.366E+00
	STD	2.867E+01	1.398E+01	1.021E+01	2.617E+01	3.688E+01	9.431E−01
f_{10}	Best	1.938E−01	4.291E−01	2.039E−05	6.449E−03	6.019E−03	9.625E−07
	Worst	2.617E+00	1.128E+00	6.342E−01	1.391E+00	2.726E+00	3.029E−01
	Mean	1.436E+00	7.718E−01	4.213E−02	4.553E−01	3.928E−01	7.635E−04
	STD	3.201E+01	1.199E+01	1.021E+01	1.811E+01	1.761E+01	1.736E−01
f_{11}	Best	1.901E−01	3.625E−01	2.526E−06	6.625E−05	3.827E−04	3.314E−08
	Worst	2.524E+00	1.524E+00	5.929E−02	1.716E−01	5.837E−01	3.412E−02
	Mean	1.625E+00	2.882E−01	4.435E−03	4.425E−02	1.888E−02	4.635E−05
	STD	1.716E+01	1.209E+01	1.033E+00	1.101E+00	1.314E+00	1.892E−02
f_{12}	Best	1.837E−01	2.736E−03	6.938E−08	2.982E−06	2.981E−03	5.928E−12
	Worst	5.029E+00	6.837E−01	2.524E−03	2.019E−02	5.928E−01	4.982E−06
	Mean	2.928E−01	2.938E−02	5.872E−05	6.042E−04	7.837E−02	2.413E−09
	STD	2.109E+01	3.029E−01	1.109E−02	3.324E−02	2.514E−02	9.209E−03

Table 2 (continued)

No.	Algorithms indices	GA	PSO	GSA	ABC	HBMO	GWO
f_{13}	Best	2.938E−04	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	5.726E−01	7.837E−01	3.826E−02	5.827E−03	4.726E−03	4.827E−03
	Mean	2.411E−02	4.726E−05	8.736E−06	9.635E−05	6.837E−04	9.736E−06
	STD	5.726E−02	4.711E−03	3.625E−03	2.313E−04	4.928E−03	4.524E−03
f_{14}	Best	3.726E−01	3.012E−01	2.932E−01	2.942E−01	2.932E−01	2.926E−01
	Worst	5.652E−01	4.625E−01	2.945E−01	2.968E−01	2.945E−01	3.092E−01
	Mean	4.635E−01	3.423E−01	2.933E−01	2.943E−01	2.933E−01	2.932E−01
	STD	7.736E−01	7.625E−01	2.938E−03	2.029E−03	2.938E−03	2.564E−04
f_{15}	Best	3.029E+00	6.736E−03	8.635E−08	7.837E−07	7.645E−06	3.263E−09
	Worst	5.827E+01	3.625E−01	2.736E−04	5.653E−02	3.526E−02	3.625E−03
	Mean	3.948E+00	8.736E−02	9.132E−06	9.736E−04	4.546E−04	6.746E−05
	STD	4.254E−01	2.413E−02	3.243E−03	2.514E−02	2.019E−03	8.764E−04
f_{16}	Best	− 1.814E+02	− 1.852E+02	− 1.861E+02	− 1.859E+02	− 1.855E+02	**− 1.867E+02**
	Worst	− 1.543E+02	− 1.452E+02	− 1.642E+02	− 1.655E+02	− 1.534E+02	− 1.645E+02
	Mean	− 1.742E+02	− 1.832E+02	− 1.749E+02	− 1.801E+02	− 1.789E+02	− 1.835E+02
	STD	2.154E+01	1.756E−01	2.635E−02	3.432E−02	1.423E−03	6.534E−03
f_{17}	Best	7.837E−12	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	3.524E−04	2.615E−03	3.625E−07	6.736E−02	3.625E−04	**0.000E+00**
	Mean	9.423E−07	4.726E−05	7.827E−08	5.554E−09	7.837E−11	**0.000E+00**
	STD	4.625E−02	2.635E−03	2.413E−04	3.524E−04	3.625E−04	**0.000E+00**
f_{18}	Best	3.726E−02	7.635E−05	**0.000E+00**	4.324E−11	2.223E−09	**0.000E+00**
	Worst	8.736E+00	5.635E−01	3.765E−02	3.625E−02	1.645E−04	5.554E−07
	Mean	5.265E−01	5.564E−03	2.847E−04	5.645E−07	5.736E−08	4.746E−09
	STD	7.827E−01	2.232E−02	8.736E−02	2.514E−02	3.433E−03	2.323E−03
f_{19}	Best	2.645E−05	**0.000E+00**	**0.000E+00**	5.546E−11	1.213E−09	**0.000E+00**
	Worst	5.654E−01	4.635E−03	4.534E−02	2.712E−03	5.635E−03	2.534E−11
	Mean	3.332E−03	6.157E−07	6.465E−09	8.453E−09	1.029E−07	**0.000E+00**
	STD	1.231E−02	3.424E−02	2.323E−02	2.093E−03	3.442E−03	**0.000E+00**
f_{20}	Best	4.635E−09	**0.000E+00**	**0.000E+00**	6.764E−12	5.453E−13	**0.000E+00**
	Worst	1.024E+01	1.372E−02	**0.000E+00**	5.736E−02	3.209E−03	**0.000E+00**
	Mean	1.726E−04	5.656E−09	**0.000E+00**	6.546E−09	2.131E−07	**0.000E+00**
	STD	1.635E+01	2.221E−02	**0.000E+00**	3.323E−03	6.564E−02	**0.000E+00**
f_{21}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	3.625E−01	7.865E−02	6.661E−03	3.029E−07	2.321E−03	7.763E−08
	Mean	8.938E−04	5.456E−08	8.001E−08	6.066E−08	5.554E−09	6.726E−10
	STD	2.221E−01	1.324E−02	1.231E−03	3.928E−05	3.231E−03	3.332E−05
f_{22}	Best	4.601E−09	**0.000E+00**	**0.000E+00**	4.434E−12	**0.000E+00**	**0.000E+00**
	Worst	3.625E−01	2.310E−02	5.029E−02	1.028E−03	2.111E−05	**0.000E+00**
	Mean	7.657E−04	7.978E−06	9.007E−07	1.004E−08	2.9387E−07	**0.000E+00**
	STD	2.312E+00	2.007E−02	1.029E−05	1.151E−03	2.019E−03	**0.000E+00**
f_{23}	Best	− 2.097E+02	− 2.099E+02	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**
	Worst	− 2.088E+02	− 2.089E+02	− 2.091E+02	− 2.093E+02	− 2.094E+02	**− 2.100E+02**
	Mean	− 2.096E+02	− 2.095E+02	− 2.097E+02	− 2.096E+02	− 2.098E+02	**− 2.100E+02**
	STD	5.928E−02	3.817E−01	1.928E−06	1.931E−05	1.625E−04	**0.000E+00**
f_{24}	Best	1.117E−19	1.348E−14	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	8.404E−10	1.241E−09	4.039E−08	2.837E−8	1.002E−09	5.837E−13
	Mean	9.881E−11	8.786E−11	2.873E−09	6.983E−14	2.121E−17	3.554E−16
	STD	2.012E−10	2.362E−10	4.622E−11	8.938E−11	2.222E−12	2.413E−12

Table 2 (continued)

No.	Algorithms indices	GA	PSO	GSA	ABC	HBMO	GWO
f_{25}	Best	1.676E−16	1.612E−17	**0.000E+00**	**0.000E+00**	3.801E−19	**0.000E+00**
	Worst	4.177E−06	3.121E−08	3.726E−09	3.625E−11	1.514E−11	6.847E−19
	Mean	1.201E−06	1.223E−09	3.726E−11	3.827E−19	3.435E−16	3.625E−23
	STD	1.300E−06	1.313E−07	5.736E−09	5.637E−10	2.382E−09	6.736E−19
f_{26}	Best	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**
	Worst	− 1.912E+00	− 1.912E+00	**− 1.913E+00**	**− 1.913E+00**	− 1.912E+00	**− 1.913E+00**
	Mean	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**
	STD	2.431E−21	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{27}	Best	2.190E−01	1.644E−07	**0.000E+00**	2.514E−11	5.736E−15	**0.000E+00**
	Worst	6.154E−01	1.632E−05	5.763E−05	4.625E−05	3.625E−05	3.625E−03
	Mean	4.143E−01	1.601E−06	2.019E−10	2.873E−07	7.938E−08	6.763E−08
	STD	3.928E+00	4.837E−04	4.635E−07	3.726E−05	3.857E−05	4.625E−06
f_{28}	Best	4.625E−03	4.928E−06	**0.000E+00**	2.534E−16	2.615E−20	**0.000E+00**
	Worst	2.918E−01	2.229E−03	3.827E−012	2.615E−10	3.625E−13	4.978E−11
	Mean	5.836E−02	6.647E−05	5.837E−017	2.524E−13	2.514E−18	3.756E−15
	STD	1.928E−02	2.091E−03	3.827E−05	2.524E−05	2.222E−05	2.645E−06
f_{29}	Best	4.479E−09	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	2.736E−04	5.625E−11	**0.000E+00**	4.625E−10	2.342E−15	**0.000E+00**
	Mean	7.432E−08	4.625E−13	**0.000E+00**	1.034E−14	8.564E−21	**0.000E+00**
	STD	4.635E−03	2.001E−04	**0.000E+00**	3.654E−07	2.009E−07	**0.000E+00**
f_{30}	Best	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**
	Worst	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**
	Mean	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{31}	Best	5.632E−03	6.667E−01	3.726E−03	3.524E−04	1.423E−03	6.667E−01
	Worst	2.514E+00	6.667E−01	8.312E−01	2.514E−03	8.032E−01	6.667E−01
	Mean	2.934E+01	6.667E−01	8.837E−02	**5.625E−04**	2.534E−02	6.667E−01
	STD	1.413E+01	**0.000E+00**	3.625E−04	2.514E−06	2.001E−03	**0.000E+00**
f_{32}	Best	1.514E+00	3.625E+00	5.070E+00	**2.511E−02**	3.615E+00	2.716E+00
	Worst	5.918E+00	7.653E+00	5.742E+00	2.019E+00	8.827E+00	6.221E+00
	Mean	2.442E+00	5.029E+00	5.154 E+00	5.029E−01	4.928E+00	4.516E+00
	STD	2.764E−01	1.029E+01	1.029E+01	3.542E−01	3.524E−01	1.625E+00
f_{33}	Best	1.267E+01	2.019E+00	1.003E+00	**9.980E−01**	9.981E−01	**9.980E−01**
	Worst	1.451E+01	3.625E+00	1.562E+00	1.514E+00	1.209E+00	1.431E+00
	Mean	1.309E+01	2.617E+00	1.102E+00	1.038E+00	1.004E+00	1.007E+00
	STD	1.029E+01	1.004E+01	3.254E−01	5.625E−01	2.514E−01	2.514E−02
f_{34}	Best	− 9.999E−01	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**
	Worst	− 9.999E−01	**− 1.000E+00**	**− 1.000E+00**	− 8.604E−01	− 9.643E−01	**− 1.000E+00**
	Mean	− 9.999E−01	**− 1.000E+00**	**− 1.000E+00**	− 9.997E−01	− 9.998E−01	**− 1.000E+00**
	STD	**9.980E−01**	**0.000E+00**	**0.000E+00**	3.726E−05	2.709E−06	**0.000E+00**
f_{35}	Best	− 4.633E+00	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**
	Worst	− 3.694E+00	**− 4.687E+00**	**− 4.687E+00**	− 5.062E+00	− 6.542E+00	**− 4.687E+00**
	Mean	− 4.372E+00	**− 4.687E+00**	**− 4.687E+00**	− 4.698E+00	− 4.732E+00	**− 4.687E+00**
	STD	1.029E+01	**0.000E+00**	**0.000E+00**	4.726E−04	5.039E−04	**0.000E+00**
f_{36}	Best	1.145E−05	**0.000E+00**	**0.000E+00**	2.423E−25	**0.000E+00**	**0.000E+00**
	Worst	5.625E−02	**0.000E+00**	**0.000E+00**	4.021E−16	1.324E−07	**0.000E+00**
	Mean	4.039E−04	**0.000E+00**	**0.000E+00**	6.039E−20	2.016E−25	**0.000E+00**
	STD	2.514E−03	**0.000E+00**	**0.000E+00**	4.423E−05	2.621E−04	**0.000E+00**

Table 2 (continued)

No.	Algorithms indices	GA	PSO	GSA	ABC	HBMO	GWO
f_{37}	Best	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**
	Worst	3.991E−01	3.979E−01	**3.978E−01**	3.981E−01	**3.978E−01**	**3.978E−01**
	Mean	3.979E−01	**3.978E−01**	**3.978E−01**	3.979E−01	**3.978E−01**	**3.978E−01**
	STD	3.625E−08	**0.000E+00**	**0.000E+00**	7.938E−12	**0.000E+00**	**0.000E+00**
f_{38}	Best	6.905E−05	3.176E−11	4.827E−18	3.029E−15	**0.000E+00**	1.524E−14
	Worst	2.275E−01	7.873E+00	3.827E−05	4.625E−05	5.827E−09	2.716E−05
	Mean	2.971E−02	1.357E+00	3.726E−10	1.425E−11	2.635E−10	2.465E−10
	STD	4.558E−02	2.262E+00	2.645E−06	2.625E−05	2.514E−07	2.091E−05
f_{39}	Best	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**
	Worst	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**
	Mean	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{40}	Best	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**
	Worst	3.151E+00	3.001E+00	3.013E+00	**3.000E+00**	**3.000E+00**	**3.000E+00**
	Mean	3.001E+00	**3.000E+00**	3.001E+00	**3.000E+00**	**3.000E+00**	**3.000E+00**
	STD	2.716E−11	**0.000E+00**	4.736E−12	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{41}	Best	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**
	Worst	− 3.789E+00	− 3.859E+00	− 3.861E+00	**− 3.862E+00**	− 3.861E+00	**− 3.862E+00**
	Mean	− 3.806E+00	− 3.861E+00	− 3.861E+00	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**
	STD	2.615E−06	4.534E−07	5.938E−08	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{42}	Best	**− 3.134E+00**	**− 3.134E+00**	**− 3.134E+00**	**− 3.134E+00**	**− 3.134E+00**	**− 3.134E+00**
	Worst	− 2.895E+00	− 2.896E+00	− 2.896E+00	− 3.109E+00	− 3.129E+00	− 3.132E+00
	Mean	− 3.126E+00	− 3.103E+00	− 3.129E+00	− 3.132E+00	− 3.132E+00	**− 3.134E+00**
	STD	2.816E−05	5.039E−04	7.635E−05	2.635E−05	5.747E−09	**0.000E+00**
f_{43}	Best	− 3.040E+00	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**
	Worst	− 2.975E+00	− 2.981E+00	− 3.039E+00	− 3.041E+00	− 3.040E+00	**− 3.042E+00**
	Mean	− 2.971E+00	− 2.998E+00	− 3.041E+00	− 3.041E+00	− 3.041E+00	**− 3.042E+00**
	STD	2.918E−01	2.736E−06	7.534E−09	6.280E−016	3.726E−10	**0.000E+00**
f_{44}	Best	1.124E−01	6.386E−04	5.837E−03	4.726E−02	5.726E−02	3.928E−03
	Worst	12.736E+00	4.796E−02	5.736E−01	6.736E−01	3.019E−01	2.615E−02
	Mean	5.733E+00	4.982E−02	8.938E−02	2.524E−01	5.029E−02	6.379E−03
	STD	2.726E+01	2.918E−02	2.635E−01	1.726E−02	1.029E+00	4.524E−02
f_{45}	Best	2.514E−03	2.889E−07	1.483E−04	5.645E−05	1.645E−05	2.213E−06
	Worst	7.201E−01	2.918E−04	5.635E−03	1.423E−03	2.098E−03	3.029E−03
	Mean	1.057E−01	2.635E−05	2.803E−04	3.625E−04	1.113E−04	5.546E−05
	STD	6.726E+00	2.413E−03	2.615E−03	8.736E−3	8.758E−03	2.098E−03
f_{46}	Best	− 5.125E+00	**− 10.536E+00**	− 5.698E+00	**− 10.536E+00**	**− 10.536E+00**	**− 10.536E+00**
	Worst	− 5.113E+00	**− 10.536E+00**	− 4.069E+00	− 10.494E+00	− 10.512E+00	− 10.531E+00
	Mean	− 5.121E+00	**− 10.536E+00**	− 4.648E+00	− 10.512E+00	− 10.530E+00	− 10.535E+00
	STD	2.413E−01	**0.000E+00**	2.837E−02	2.716E−03	2.615E−05	2.756E−11
f_{47}	Best	− 305.01E+00	− 335.76E+00	**− 391.66E+00**	**− 391.66E+00**	**− 391.66E+00**	**− 391.66E+00**
	Worst	− 248.11E+00	− 320.97E+00	− 384.53E+00	− 388.31E+00	− 387.32E+00	− 390.82E+00
	Mean	− 278.34E+00	− 349.25E+00	− 391.09E+00	− 389.54E+00	− 390.02E+00	− 390.52E+00
	STD	1.857E+01	1.524E+01	3.625E−02	2.476E−02	7.635E−03	5.487E−03
f_{48}	Best	1.534E−01	3.654E−02	5.746E−02	1.435E−03	2.504E−03	7.635E−04
	Worst	1.018E+00	2.531E−01	1.524E−01	2.546E−01	1.524E−02	2.029E−02
	Mean	2.524E−01	4.536E−02	6.423E−02	8.093E−02	5.762E−03	1.635E−03
	STD	9.645E−01	2.736E−03	3.029E−03	1.221E−02	3.524E−03	5.645E−03

Table 2 (continued)

No.	Algorithms indices	GA	PSO	GSA	ABC	HBMO	GWO
f_{49}	Best	3.625E−19	**0.000E+00**	4.635E−32	**0.000E+00**	2.615E−32	**0.000E+00**
	Worst	2.534E−07	**0.000E+00**	2.813E−23	4.736E−21	2.615E−21	1.726E−16
	Mean	3.625E−013	**0.000E+00**	5.837E−29	2.736E−29	7.673E−29	5.837E−21
	STD	2.413E−01	**0.000E+00**	2.514E−02	2.514E−03	2.514E−04	2.011E−05
f_{50}	Best	− 41.524E+00	− 42.903E+00	− 42.668E+00	− 42.587E+00	− 42.645E+00	− 42.705E+00
	Worst	− 40.524E+00	− 41.885E+00	− 41.702E+00	− 41.324E+00	− 41.785E+00	− 41.524E+00
	Mean	− 41.021E+00	− 42.562E+00	− 42.143E+00	− 42.611E+00	− 42.201E+00	− 41.998E+00
	STD	2.514E−01	1.425E−02	2.993E−02	2.736E−02	5.546E−02	4.524E−02
f_{51}	Best	3.625E−08	**0.000E+00**	**0.000E+00**	2.837E−11	2.534E−10	**0.000E+00**
	Worst	2.514E−05	**0.000E+00**	3.524E−08	2.514E−07	7.635E−05	**0.000E+00**
	Mean	3.625E−06	**0.000E+00**	5.216E−10	8.756E−09	3.425E−09	**0.000E+00**
	STD	3.625E−02	**0.000E+00**	4.938E−04	3.524E−03	6.744E−04	**0.000E+00**
f_{52}	Best	2.413E−09	2.511E−13	2.514E−10	9.534E−12	2.415E−15	2.413E−16
	Worst	4.736E−04	4.736E−08	4.736E−07	2.514E−05	9.746E−10	4.938E−07
	Mean	3.827E−08	4.625E−10	3.332E−09	2.847E−09	2.231E−13	2.756E−10
	STD	3.324E−02	4.635E−06	2.091E−04	2.514E−03	2.012E−04	2.003E−05
f_{53}	Best	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**
	Worst	− 387.52E+00	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	− 395.63E+00	− 397.63E+00
	Mean	− 398.43E+00	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	− 399.78E+00	− 399.53E+00
	STD	3.827E−10	**0.000E+00**	**0.000E+00**	**0.000E+00**	2.645E−12	5.645E−13

No.	Algorithms indices	COA	LCA	HGSO	SSA	BSSA
f_1	Best	7.873E−06	9.542E−07	4.635E−16	2.241E−10	**8.881E−018**
	Worst	5.563E−02	0.654E−01	5.736E−12	2.586E−03	**9.763E−017**
	Mean	2.098E−04	6.763E−03	3.928E−15	1.391E−04	**8.867E−018**
	STD	9.340E−02	9.653E−03	2.351E−03	4.851E−04	**2.113E−04**
f_2	Best	5.127E−02	9.546E−03	8.625E−06	9.645E−05	**1.827E−07**
	Worst	3.428E+00	5.873E−02	3.524E−04	5.093E−03	**4.726E−07**
	Mean	1.090E−01	6.453E−03	2.872E−05	8.878E−05	**3.092E−07**
	STD	7.893E+00	1.092E−01	2.651E−03	1.984E−02	**5.928E−05**
f_3	Best	− 1.867E+00	− 1.999E+00	**− 2.062E+00**	**− 2.062E+00**	**− 2.062E+00**
	Worst	− 1.023E+00	− 1.072E+00	− 1.435E+00	− 1.675E+00	**− 1.894E+00**
	Mean	− 1.648E+00	− 1.864E+00	− 1.998E+00	− 1.994E+00	**− 2.001E+00**
	STD	3.265E−02	5.435E−03	2.716E−03	2.565E−03	**3.625E−04**
f_4	Best	− 9.763E−01	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**
	Worst	− 1.039E−01	− 8.958E−01	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**
	Mean	− 6.543E−01	− 9.876E−01	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**
	STD	2.327E−01	2.153E−05	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_5	Best	− 9.564E+02	− 9.575E+02	**− 9.596E+02**	**− 9.596E+02**	**− 9.596E+02**
	Worst	− 9.323E+02	− 9.422E+02	− 9.512E+02	− 9.528E+02	**− 9.572E+02**
	Mean	− 9.488E+02	− 9.511E+02	− 9.541E+02	− 9.543E+02	**− 9.589E+02**
	STD	2.114E−04	5.664E−04	1.928E−03	2.182E−03	**3.094E−04**
f_6	Best	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**
	Worst	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**
	Mean	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**	**− 8.690E−01**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_7	Best	2.897E−08	9.878E−08	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	8.675E−01	2.546E−03	3.726E−04	4.137E−05	**0.000E+00**
	Mean	2.653E−04	6.938E−06	2.657E−06	3.435E−06	**0.000E+00**
	STD	3.332E−05	2.635E−05	2.837E−03	9.670E−06	**0.000E+00**

Table 2 (continued)

No.	Algorithms indices	COA	LCA	HGSO	SSA	BSSA
f_8	Best	− 19.001E+00	− 18.672E+00	− 19.205E+00	− 19.102E+00	**− 19.208E+00**
	Worst	− 12.817E+00	− 10.527E+00	− 18.786E+00	− 17.572E+00	**− 18.895E+00**
	Mean	− 18.897E+00	− 18.562E+00	− 19.132E+00	− 18.889E+00	**− 19.172E+00**
	STD	7.038E+00	2.109E−01	3.092E−03	5.978E−03	**2.019E−03**
f_9	Best	**− 1.400E+00**	− 1.399E+00	**− 1.400E+00**	− 1.397E+00	**− 1.400E+00**
	Worst	− 1.119E+00	− 9.998E−01	− 1.176E+00	− 1.122E+00	**− 1.270E+00**
	Mean	− 1.289E+00	− 1.332E+00	− 1.321E+00	− 1.302E+00	**− 1.393E+00**
	STD	1.290E+01	1.105E+00	4.514E−02	1.233E−01	**4.302E−02**
f_{10}	Best	5.265E−03	7.836E−06	3.726E−08	2.018E−07	**0.000E+00**
	Worst	7.827E+00	2.918E−01	2.615E−02	4.209E−03	**8.376E−05**
	Mean	3.726E−02	1.524E−03	7.312E−04	1.534E−04	**2.726E−07**
	STD	2.015E+01	3.948E−01	3.029E−02	3.029E−02	**2.938E−02**
f_{11}	Best	5.723E−05	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	4.524E−01	4.982E−02	2.716E−03	4.791E−02	**2.111E−06**
	Mean	3.019E−02	2.425E−05	9.645E−06	1.998E−05	**3.762E−08**
	STD	2.325E−01	1.029E−02	2.514E−02	1.155E−02	**1.067E−02**
f_{12}	Best	5.928E−08	6.736E−06	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	4.635E−02	4.625E−02	5.736E−08	7.665E−06	**0.000E+00**
	Mean	1.029E−04	7.625E−04	2.716E−10	4.905E−07	**0.000E+00**
	STD	2.534E−02	3.726E−03	2.514E−05	1.505E−06	**0.000E+00**
f_{13}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	6.736E−03	8.736E−06	3.524E−08	9.715E−03	**1.726E−18**
	Mean	8.736E−07	3.625E−08	2.817E−09	9.715E−04	**0.000E+00**
	STD	2.323E−03	2.165E−03	1.546E−05	2.964E−03	**0.000E+00**
f_{14}	Best	2.929E−01	2.928E−01	**2.925E−01**	**2.925E−01**	**2.925E−01**
	Worst	2.974E−01	3.019E−01	2.997E−01	2.998E−01	**2.956E−01**
	Mean	2.951E−01	2.943E−01	2.945E−01	2.967E−01	**2.931E−01**
	STD	3.524E−04	6.736E−03	1.022E−03	3.029E−03	**2.091E−04**
f_{15}	Best	7.837E−06	7.354E−05	4.423E−09	2.265E−08	**6.524E−11**
	Worst	2.635E−01	3.524E−02	3.021E−05	7.122E−03	**5.625E−08**
	Mean	3.928E−03	8.635E−04	2.615E−07	5.181E−04	**3.265E−10**
	STD	1.245E−02	2.534E−04	2.451E−03	1.412E−03	**1.524E−04**
f_{16}	Best	− 1.866E+02	− 1.865E+02	**− 1.867E+02**	− 1.866E+02	**− 1.867E+02**
	Worst	− 1.657E+02	− 1.645E+02	− 1.786E+02	− 1.799E+02	**− 1.803E+02**
	Mean	− 1.854E+02	− 1.833E+02	− 1.811E+02	**− 1.824E+02**	− 1.823E+02
	STD	3.524E−03	5.534E−03	1.563E−03	3.625E−04	**2.736E−04**
f_{17}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	9.635E−08	5.726E−08	**0.000E+00**	2.514E−14	**0.000E+00**
	Mean	2.817E−13	2.415E−11	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	1.635E−05	2.938E−05	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{18}	Best	6.657E−12	2.423E−13	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	4.536E−06	7.536E−05	2.645E−07	7.857E−06	**8.867E−09**
	Mean	2.534E−09	6.665E−07	2.453E−10	8.887E−09	**5.453E−11**
	STD	4.443E−04	3.343E−04	2.720E−04	4.534E−04	**2.534E−04**
f_{19}	Best	5.736E−09	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	4.635E−04	3.625E−06	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Mean	7.635E−06	3.736E−09	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	2.514E−05	2.847E−03	**0.000E+00**	**0.000E+00**	**0.000E+00**

Table 2 (continued)

No.	Algorithms indices	COA	LCA	HGSO	SSA	BSSA
f_{20}	Best	8.887E−13	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	5.635E−04	6.675E−03	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Mean	4.535E−09	2.434E−07	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	3.434E−03	9.978E−04	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{21}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	2.413E−03	5.736E−06	2.643E−07	4.837E−06	**5.635E−16**
	Mean	2.918E−09	3.534E−10	2.534E−12	8.726E−11	**0.000E+00**
	STD	5.736E−03	7.938E−04	2.564E−05	2.524E−04	**0.000E+00**
f_{22}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	4.029E−02	1.524E−03	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Mean	2.019E−06	5.298E−06	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	3.625E−02	1.524E−03	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{23}	Best	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**
	Worst	− 2.095E+02	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**
	Mean	− 2.097E+02	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**	**− 2.100E+02**
	STD	1.012E−05	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{24}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	1.262E−29	**0.000E+00**
	Worst	2.716E−11	1.524E−19	3.562E−12	**8.125E−24**	5.736E−19
	Mean	4.625E−19	2.726E−20	9.563E−21	9.585E−25	**0.000E+00**
	STD	3.524E−09	3.522E−13	2.039E−15	1.799E−24	**0.000E+00**
f_{25}	Best	4.736E−24	**0.000E+00**	**0.000E+00**	1.511E−29	**0.000E+00**
	Worst	1.928E−15	3.928E−11	2.538E−13	2.070E−24	**0.000E+00**
	Mean	2.342E−21	4.837E−19	1.034E−19	1.542E−25	**0.000E+00**
	STD	4.938E−10	6.847E−08	6.741E−6	4.757E−9	**0.000E+00**
f_{26}	Best	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**
	Worst	− 1.912E+00	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**
	Mean	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**	**− 1.913E+00**
	STD	2.723E−27	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{27}	Best	4.039E−13	7.635E−15	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	7.923E−04	2.745E−06	2.019E−07	2.615E−08	**3.524E−11**
	Mean	2.165E−10	3.029E−10	4.536E−09	2.635E−11	**3.726E−17**
	STD	4.928E−06	4.672E−06	1.435E−05	6.736E−06	**3.948E−07**
f_{28}	Best	5.736E−15	**0.000E+00**	**0.000E+00**	1.995E−23	**0.000E+00**
	Worst	3.524E−08	3.625E−11	2.546E−09	1.522E−07	**0.000E+00**
	Mean	1.029E−10	2.938E−17	1.546E−11	5.221E−09	**0.000E+00**
	STD	2.021E−06	3.625E−06	3.854E−07	2.777E−08	**0.000E+00**
f_{29}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	5.098E−13	2.937E−13	2.534E−15	1.029E−13	**0.000E+00**
	Mean	6.097E−16	2.645E−19	1.763E−19	2.413E−21	**0.000E+00**
	STD	7.321E−06	3.827E−06	3.542E−07	4.403E−07	**0.000E+00**
f_{30}	Best	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**
	Worst	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**
	Mean	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**	**− 1.031E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{31}	Best	2.413E−01	2.514E−01	1.385E−03	6.667E−01	**1.423E−04**
	Worst	4.625E+00	7.534E−01	3.545E−02	6.667E−01	**1.524E−03**
	Mean	6.524E−01	3.334E−01	2.091E−03	6.667E−01	6.524E−04
	STD	1.928E−01	2.514E−04	5.653E−05	**0.000E+00**	1.423E−06

Table 2 (continued)

No.	Algorithms indices	COA	LCA	HGSO	SSA	BSSA
f_{32}	Best	1.625E−01	7.615E−01	2.896E−02	3.524E+00	2.514E−02
	Worst	7.514E+00	4.019E+00	2.925E+00	5.387E+00	**1.029E+00**
	Mean	4.827E+00	9.837E−01	5.402E−01	4.937E+00	**4.726E−01**
	STD	2.615E−01	2.514E−02	5.732E−02	2.534E−02	**1.635E−02**
f_{33}	Best	9.984E−01	**9.980E−01**	**9.980E−01**	**9.980E−01**	**9.980E−01**
	Worst	1.201E+00	1.052E+00	2.635E+00	1.429E+00	**1.002E+00**
	Mean	1.092E+00	9.985E−01	7.073E+00	1.023E+00	**9.984E−01**
	STD	4.039E−01	4.029E−03	5.099E−03	5.039E−03	**3.827E−03**
f_{34}	Best	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**
	Worst	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**
	Mean	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**	**− 1.000E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{35}	Best	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**
	Worst	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**
	Mean	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**	**− 4.687E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{36}	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	2.615E−06	2.039E−07	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Mean	2.514E−14	2.016E−15	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	2.510E−05	5.028E−05	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{37}	Best	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**
	Worst	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**
	Mean	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**	**3.978E−01**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{38}	Best	1.524E−15	1.342E−14	**0.000E+00**	8.556E−21	**0.000E+00**
	Worst	2.413E−07	2.019E−08	5.641E−05	2.487E−08	**2.314E−09**
	Mean	5.625E−11	5.635E−11	2.003E−10	1.430E−09	**5.435E−12**
	STD	2.635E−06	2.645E−05	4.536E−06	**4.690E−09**	2.837E−07
f_{39}	Best	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**
	Worst	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**
	Mean	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**	**− 6.020E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{40}	Best	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**
	Worst	**3.000E+00**	3.002E+00	**3.000E+00**	**3.000E+00**	**3.000E+00**
	Mean	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{41}	Best	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**
	Worst	**− 3.862E+00**	− 3.860E+00	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**
	Mean	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**	**− 3.862E+00**
	STD	**0.000E+00**	1.625E−13	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{42}	Best	**− 3.134E+00**	**− 3.134E+00**	**− 3.134E+00**	**− 3.134E+00**	**− 3.134E+00**
	Worst	− 3.124E+00	− 3.103E+00	− 3.003E+00	− 3.109E+00	**− 3.134E+00**
	Mean	− 3.133E+00	− 3.132E+00	− 3.075E+00	− 3.132E+00	**− 3.134E+00**
	STD	5.039E−08	4.093E−08	2.642E−08	4.827E−11	**0.000E+00**
f_{43}	Best	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**
	Worst	− 3.028E+00	− 3.018E+00	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**
	Mean	− 3.041E+00	− 3.039E+00	**− 3.042E+00**	**− 3.042E+00**	**− 3.042E+00**
	STD	2.514E−06	2.413E−10	**0.000E+00**	**0.000E+00**	**0.000E+00**

Table 2 (continued)

No.	Algorithms indices	COA	LCA	HGSO	SSA	BSSA
f_{44}	Best	2.134E−02	6.423E−02	6.542E−03	4.514E−03	**2.039E−04**
	Worst	1.029E+00	9.413E−01	9.736E−02	2.813E−01	**6.423E−03**
	Mean	2.904E−01	8.524E−02	7.534E−03	4.534E−02	**3.023E−03**
	STD	1.025E−01	**2.423E−03**	3.096E−03	9.413E−03	2.514E−03
f_{45}	Best	1.079E−07	1.423E−04	**0.000E+00**	2.843E−04	**0.000E+00**
	Worst	2.534E−05	8.938E−03	5.837E−04	4.625E−02	**0.000E+00**
	Mean	6.675E−06	2.054E−04	2.645E−08	3.477E−04	**0.000E+00**
	STD	4.036E−05	2.514E−05	2.534−03	6.543E−03	**0.000E+00**
f_{46}	Best	**− 10.536E+00**	− 10.535E+00	**− 10.536E+00**	**− 10.536E+00**	**− 10.536E+00**
	Worst	− 10.413E+00	− 10.281E+00	− 10.511E+00	− 10.534E+00	**− 10.536E+00**
	Mean	− 10.511E+00	− 10.530E+00	− 10.532E+00	− 10.535E+00	**− 10.536E+00**
	STD	5.736E−02	2.514E−04	2.546E−06	3.524E−08	**0.000E+00**
f_{47}	Best	**− 391.66E+00**	**− 391.66E+00**	**− 391.66E+00**	**− 391.66E+00**	**− 391.66E+00**
	Worst	− 386.53E+00	− 359.01E+00	− 360.64E+00	− 349.25E+00	**− 387.76E+00**
	Mean	− 387.51E+00	− 388.8E+00	− 387.64E+00	− 363.38E+00	**− 390.14E+00**
	STD	6.857E−03	8.645E−03	3.922E−02	1.413E−01	**3.514E−03**
f_{48}	Best	2.534E−05	2.501E−04	9.645E−05	4.635E−06	**2.564E−07**
	Worst	2.029E−03	5.635E−03	1.423E−03	3.524E−04	**6.039E−06**
	Mean	3.762E−04	2.817E−04	8.867E−04	1.509E−05	**3.748E−07**
	STD	2.635E−03	2.514E−04	2.019E−03	3.625E−04	**2.342E−04**
f_{49}	Best	**0.000E+00**	5.625E−48	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	6.746E−15	7.736E−17	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Mean	3.746E−25	5.872E−37	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	2.938E−04	2.625E−04	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{50}	Best	− 42.90E+00	− 42.901E+00	− 42.932E+00	− 42.991E+00	− 42.991E+00
	Worst	− 42.001E+00	− 41.029E+00	− 41.524E+00	− 41.625E+00	− 42.923E+00
	Mean	− 42.425E+00	− 42.221E+00	− 42.901E+00	− 42.091E+00	− 42.965E+00
	STD	2.534E−02	3.021E−02	3.625E−02	2.534E−02	**2.541E−03**
f_{51}	Best	2.514E−13	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Worst	5.736E−08	2.938E−09	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Mean	8.746E−11	3.726E−12	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	3.049E−03	3.625E−03	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{52}	Best	2.534E−32	5.746E−23	4.635E−182	3.625E−13	**0.000E+00**
	Worst	5.376E−15	3.928E−17	**4.736E−133**	3.726E−09	3.726E−129
	Mean	8.293E−30	3.265E−21	2.736E−156	6.756E−12	**6.534E−199**
	STD	3.625E−05	6.847E−06	**0.000E+00**	3.948E−06	**0.000E+00**
f_{53}	Best	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**
	Worst	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**
	Mean	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**	**− 400.00E+00**
	STD	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**

Best, Worst, Mean and STD denotes the best solution, the worst solution, the mean solution and the standard deviation, respectively

proposed optimization algorithm has strong consistency in accuracy as proved by the ANOVA.

To sum up, based on the aforementioned graphical results, convergence and ANOVA tests, it can be easily obvious that the proposed BSSA shows superior performance compared to other optimization algorithms in solving different benchmark functions.

In other side, the meta-heuristic optimization methods aim to balance the exploitation and exploration terms to use all capability in search process. Exploitation used a kind of upgraded operator having better solution to improve the convergence. The exploration term creates the new solutions through the search space to keep away from local solutions. To calculate the exploration and the

Table 3 Average error rates presented by employed algorithms, for some of benchmark functions

No.	GA	PSO	GSA	ABC	HBMO	GWO	COA	LCA	HGSO	SSA	BSSA
f_1	2.315E−02	6.454E−03	2.452E−03	1.024E−05	2.451E−04	5.464E−04	5.641E−05	2.201E−05	1.564E−06	5.264E−05	**1.0657E−07**
f_6	5.164E−07	3.054E−08	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{11}	5.142E−01	8.145E−04	4.015E−03	4.152E−03	4.054E−06	4.054E−05	5.046E−04	5.648E−06	4.120E−06	3.148E−06	**1.024E−06**
f_{16}	1.021E−04	1.087E−05	2.054E−04	6.045E−05	7.045E−05	4.054E−07	5.045E−06	4.024E−06	2.014E−08	4.052E−07	**0.000E+00**
f_{21}	4.150E−07	4.054E−12	1.121E−13	8.045E−14	4.054E−14	5.014E−16	4.051E−18	2.014E−13	5.965E−20	5.457E−18	**1.021E−23**
f_{26}	5.187E−24	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	4.154E−42	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{31}	2.014E−05	1.584E−04	5.147E−04	3.054E−05	7.154E−05	4.054E−05	8.786E−06	7.184E−07	4.157E−10	4.154E−10	7.154E−16
f_{36}	4.519E−15	**0.000E+00**	**0.000E+00**	7.485E−17	4.517E−11	1.043E−14	**0.000E+00**	5.152E−15	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{41}	6.584E−20	4.325E−24	5.654E−25	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
f_{46}	6.596E−15	**0.000E+00**	8.574E−19	6.857E−22	6.354E−15	8.564E−15	5.426E−14	4.521E−19	4.021E−22	6.254E−26	**0.000E+00**
f_{51}	9.665E−10	**0.000E+00**	2.895E−14	3.654E−17	5.698E−15	**0.000E+00**	6.354E−16	5.684E−18	**0.000E+00**	**0.000E+00**	**0.000E+00**

Table 4 Rank of algorithms for some of selected benchmark functions using MAE

Algorithm	MAE	Rank
BSSA	**0.0027**	**1**
HGSO	0.0029	2
COA	0.0338	3
LCA	0.0439	4
GWO	0.0633	5
SSA	0.0724	6
ABC	0.0755	7
HBMO	0.1122	8
PSO	0.2333	9
GSA	0.4974	10
GA	1.189	11

exploitation abilities for BSSA used similar formulation from Hashim et al. (2019). Hereby, the increased and decreased mean values of distance within dimensions of population individual show the exploration and the exploitation, respectively. The dimension-wise diversity (Div) through a cycle of search process can be expressed by (Hashim et al. 2019):

$$\frac{1}{\mathrm{Div}_j} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{median}(x^j) - x_i^j, \quad \mathrm{Div}^t = \frac{1}{N}\sum_{j=1}^{D} \mathrm{Div}_j \tag{15}$$

where x_i^j and $\mathrm{median}(x_j)$ denote the jth dimension of ith population and the median value of jth dimension of all population of size N. The supposed terms can be calculated as follows (Hashim et al. 2019):

$$\begin{aligned} \mathrm{Exploration} &= \frac{\mathrm{Div}^t}{\mathrm{Div}_{\max}} \times 100, \\ \mathrm{Exploitation} &= \frac{\left|\mathrm{Div}^t - \mathrm{Div}_{\max}\right|}{\mathrm{Div}_{\max}} \times 100 \end{aligned} \tag{16}$$

where Div^t and $\mathrm{Div}_{\max}$ denote the population diversity in tth iteration and the maximum diversity found in the entire T iterations, respectively. To analyze these terms, four common benchmarks are employed from Table 1 and simulation results are shown in Figs. 13 and 14. As seen from these figures, BSSA efficiently keeps a balance between the exploration and exploitation ratios during most of the search process.

According to optimization objective functions and efficiently search the large space, the largest possible rejection region controls the type I error rate. Affecting the closed testing attitude optimized multiple testing procedures with strong family-wise error rate control is created. Based on statistical index, the family-wise error rate (FWER) is considered as the possibility of building one or extra false discoveries between all the hypotheses when applying multiple pair-wise tests. The combination of pair-wise comparisons can be defined by (García et al. 2009):

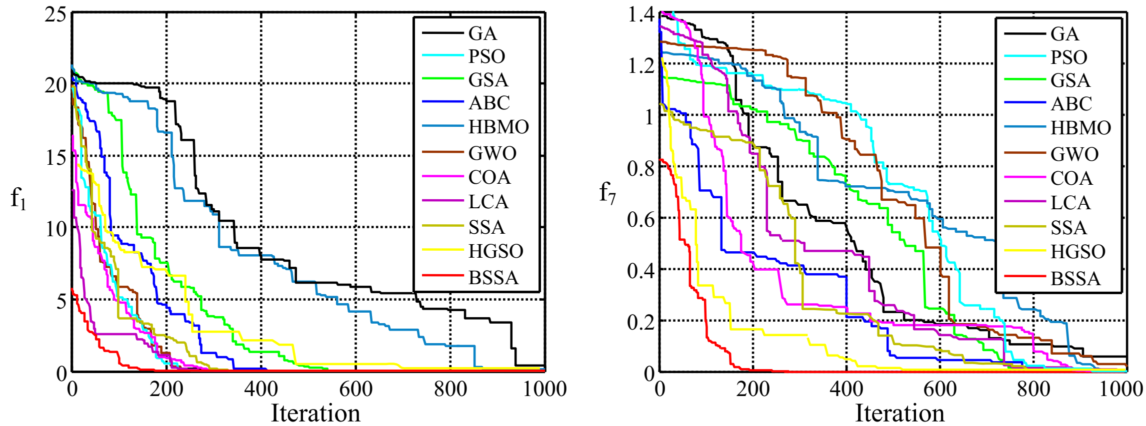

Fig. 6 Evolution rate comparison for two benchmark functions, f_1 and f_7, versus the number of function evaluations

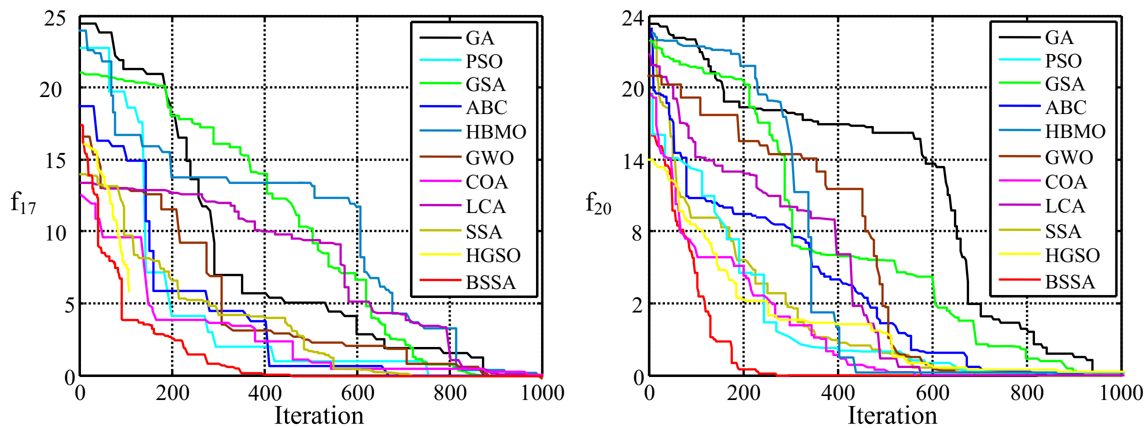

Fig. 7 Evolution rate comparison for two benchmark functions, f_{17} and f_{20}, versus the number of function evaluations

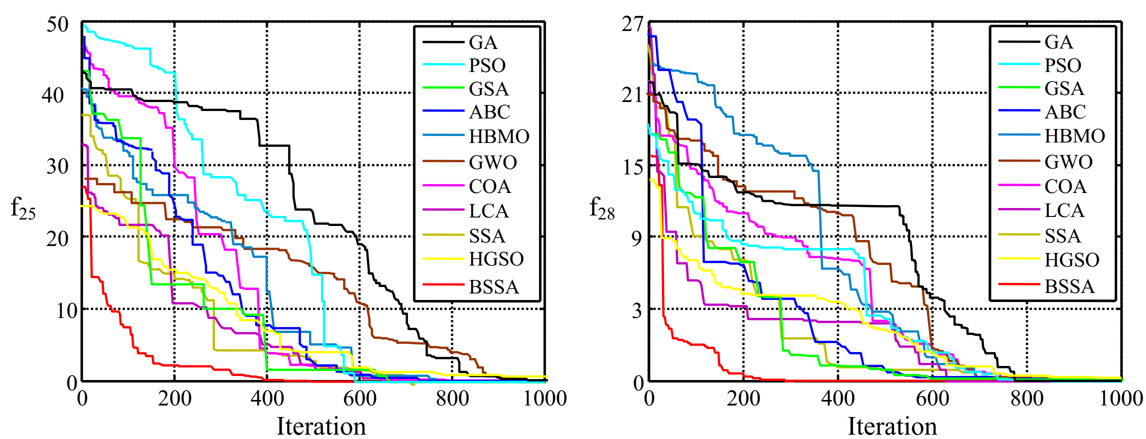

Fig. 8 Evolution rate comparison for two benchmark functions, f_{25} and f_{28}, versus the number of function evaluations

$$P = P(\text{reject}\, H_0 | H_0\, \text{true}) = 1 - P(\text{accept}\, \mathrm{H}_0 | H_0\, \text{true}). \tag{17}$$

Interested reader can refer García et al. (2009) to get more details. The previous analysis checked the behavior of the optimization algorithms, while Table 5 tabulates the differences between using multiple comparisons measures. It shows the sum of rankings calculated in each comparison and the associated p value (García et al. 2009). According to this table, BSSA actually outperforms other selected

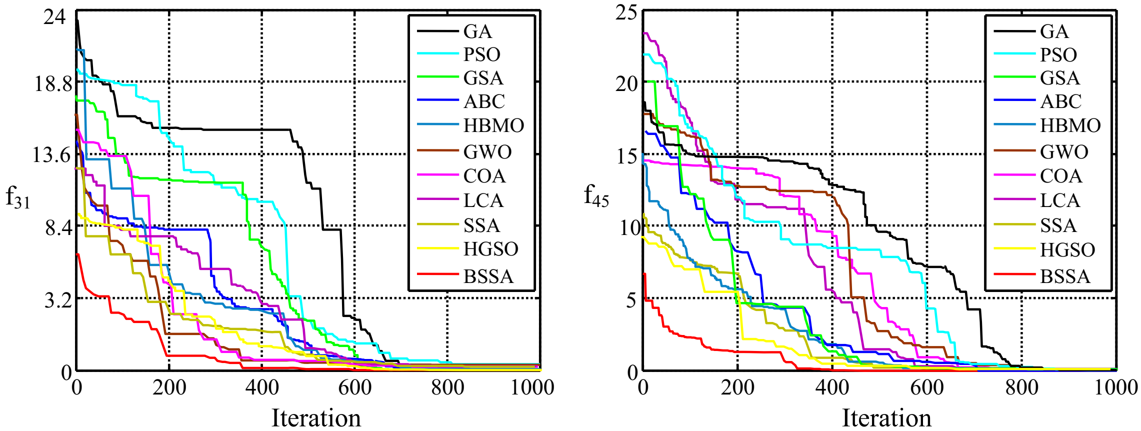

Fig. 9 Evolution rate comparison for two benchmark functions, f_{31} and f_{45}, versus the number of function evaluations

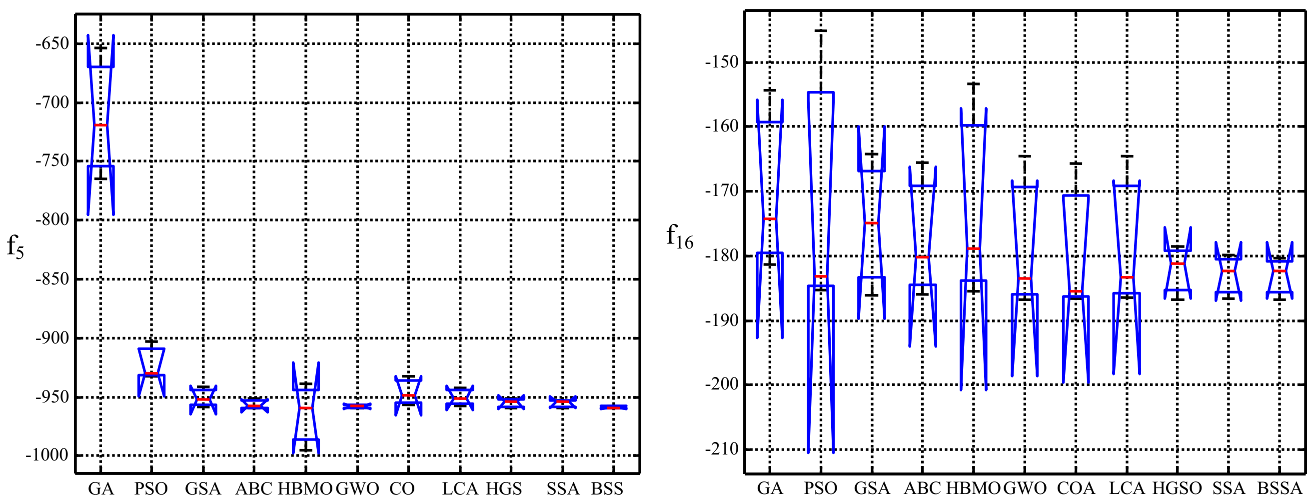

Fig. 10 The ANOVA test for all employed algorithms for two benchmark functions, f_5 and f_{31}

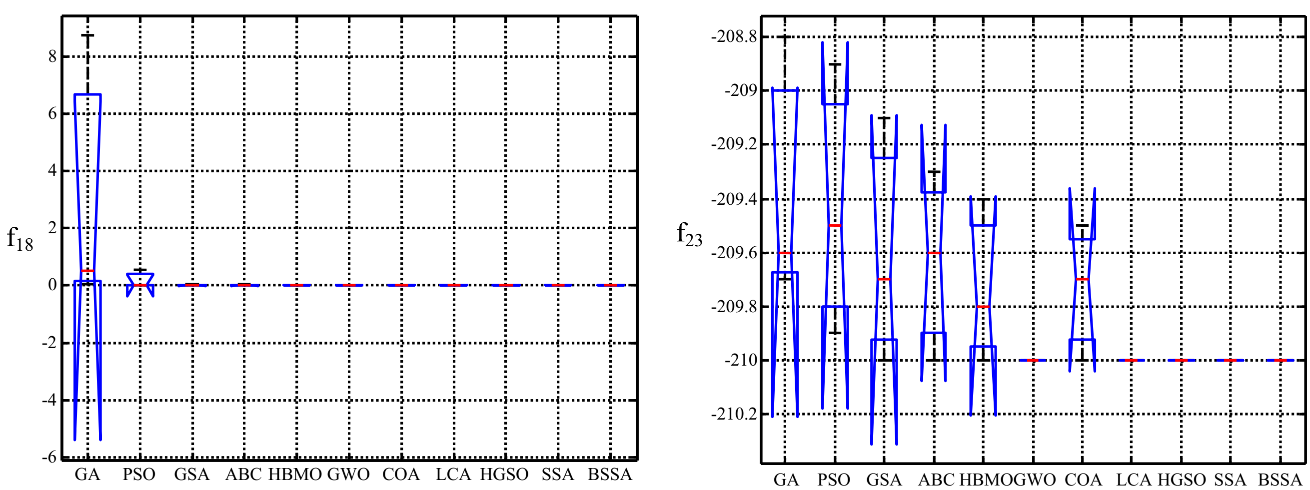

Fig. 11 The ANOVA test for all employed algorithms for two benchmark functions, f_{18} and f_{23}

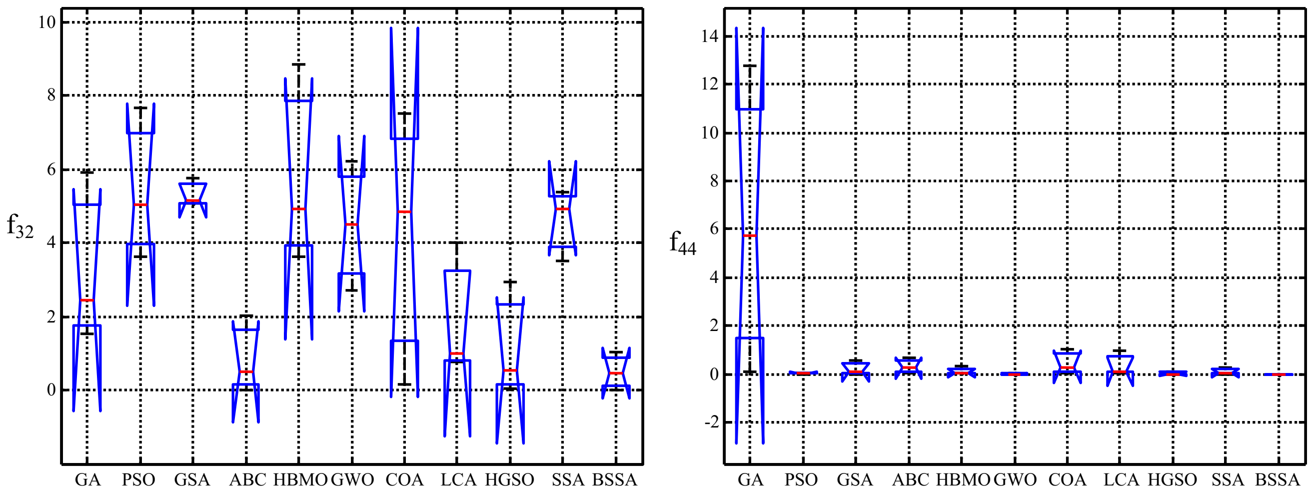

Fig. 12 The ANOVA test for all employed algorithms for two benchmark functions, f_{32} and f_{44}

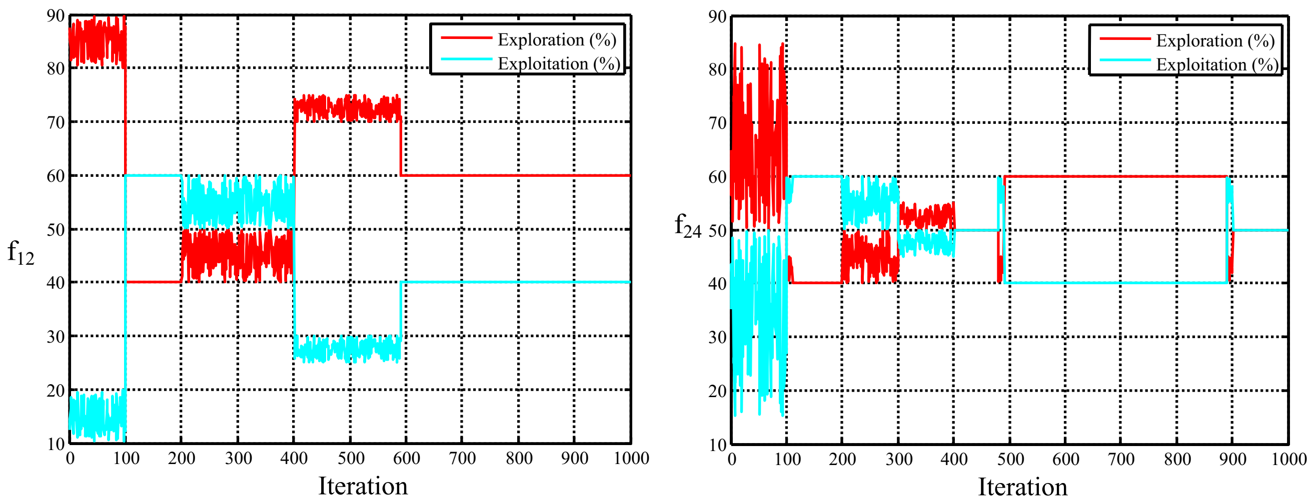

Fig. 13 Evaluation of exploration and exploitation patterns for BSSA in two benchmark functions, f_{12} and f_{24}

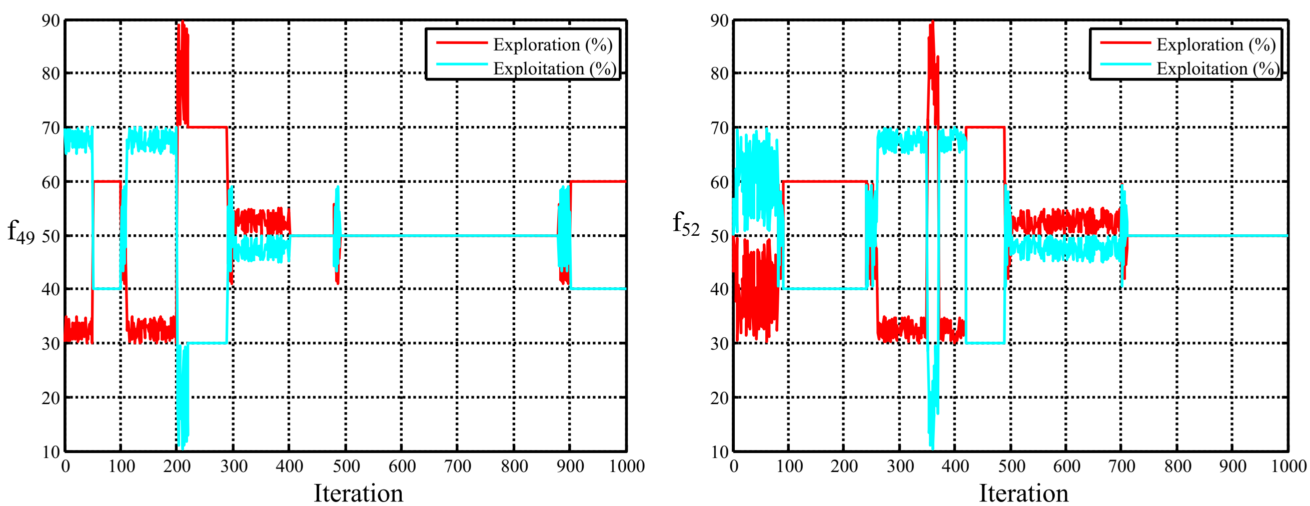

Fig. 14 Evaluation of exploration and exploitation patterns for BSSA in two benchmark functions, f_{49} and f_{52}

Table 5 Wilcoxon test considering functions f_{10}–f_{15}

BSSA versus	R^+	R^-	p value
GA	63.23	2.54	0.007
PSO	61.02	2.92	0.134
GWO	61.22	2.78	0.121
HGSO	73.62	1.13	0.006
SSA	75.93	1.02	0.005

methods taking into account independent pair-wise comparisons due to the fact that the p values are below $\alpha = 0.05$. According to Eq (14) and Table 5, it can be found BSSA with p value of $P = 1 - ((1 - 0.007) \times (1 - 0.134) \times (1 - 0.121) \times (1 - 0.006) \times (1 - 0.005)) = 0.252$ considering all objective functions. In Table 5, R^+ and R^- are the sum of ranks for the functions on which the second method outperformed the first, and R^- the sum of ranks for the opposite (García et al. 2009).

3.2 Real-world engineering problems

3.2.1 Power system stabilizer design on single machine infinite bus system

To get more evaluation of the proposed BSSA, a real-world engineering problem is employed in this section. It is optimal design of power system stability in the single machine infinite bus (SMIB) (Ghasemi et al. 2013). The final obtained result by BSSA is compared to the recent available methods in order to evaluate its optimum result.

Power system stability is a big challenge for engineering to damp low-frequency oscillation in an acceptable time after unexpected faults. Low-frequency oscillation notes that the generator rotor angle oscillations have a special frequency between 0.1 and 3.0 Hz and are defined by how they are shaped in the real-world electric power system (Shayeghi and Ghasemi 2014). Usually, automatic voltage regulators (AVRs) employed to enhance the steady-state stability of the electric power systems, but transient stability becomes a main concern for the power system operators. Therefore, a supplementary stabilizer such as power system stabilizers (PSSs) dedicated on the control loop (Shayeghi and Ghasemi 2014). As a dark point, a conventional PSS cannot guarantee the electric power system stability over a wide range of operating condition because it designed at a limit range. To tackle the shortage of the classical PSS, the optimal tuning of PSS parameters is converted an optimization problem and intelligence algorithm employed to find its parameters. Shorting speaking, the mathematical models are (Ghasemi et al. 2013):

$$\dot{\delta} = \omega - \omega_0 \tag{18}$$

$$\dot{\omega} = \frac{1}{M}\left(P_m - P_e - \frac{D(\omega - \omega_0)}{\omega_0}\right) \tag{19}$$

$$\dot{E}'_q = \frac{1}{T'_{do}}\left[E_{fd} - (x_d - x'_d)i_d - E'_q\right] \tag{20}$$

$$\dot{E}''_q = \frac{1}{T''_{do}}\left[-E_{fd} - (x'_d - x''_d)i_d - E''_q + E'_q\right] + \dot{E}'_q \tag{21}$$

$$\dot{E}''_d = \frac{1}{T''_{qo}}\left[(x_q - x''_q)i_q - E''_d\right]. \tag{22}$$

Power system exciter can be expressed as:

$$\dot{E}_{fd} = \frac{1}{T_a}\left[K_a(V_{\text{ref}} - V - U) - E_{fd}\right]. \tag{23}$$

The algebraic formulations are

$$T_e = E'_d i_d + E'_q i_q - (x'_q - x'_d)i_d i_q \tag{24}$$

$$E_d = -r_a i_d + \frac{\dot{E}''_q}{\omega_0} + E''_d + x''_q i_q \tag{25}$$

$$E_q = -r_a i_q + \frac{\dot{E}''_d}{\omega_0} + E''_q + x''_d i_d \tag{26}$$

where the subscripts q and d denote the quadrature axis (q-axis) and direct axis (d-axis) in the induction machine, respectively. Hereby, x_d, x_q, T'_{do}, T'_{qo}, δ and ω denote transient reactance, open circuit time constants, rotor angle and angular speed, respectively. P_{mi}, M_i, D_i, H_i and P_e state input mechanical power, machine inertia, damping coefficient, inertia constant and electrical power of the induction machine, respectively. E_{fd}, K_a and T_a refer to the field voltage, the excitation system gain and the time constant, respectively. The PSS typically utilizes rotor speed, bus frequency or active power output as input's signal (Shayeghi and Ghasemi 2012). As shown in Fig. 15, the classical PSS consists of two lead-lag filters. K_{pss} denotes the stabilizer gain. T_w and T_1,…, T_4 refer to the washout time constants and lead-lag filters, respectively. Input signal is speed deviation and output signal is added to the difference result between V_{ref} and V_t which it will shown early.

The employed objective function dictates power system to get the minimum 5% damping for all oscillatory modes overall operating conditions; one gets:

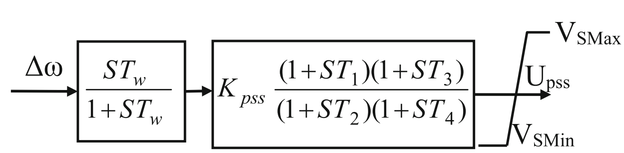

Fig. 15 Structure of the conventional power system stabilizer

$$J = \sum_{j=1}^{N_p} \sum_{i=1}^{N_g} \max_{i,j} [\mathrm{Re}(\lambda_{i,j}) - \min\{-\zeta |\mathrm{Im}(\lambda_{i,j})| \times \beta\}] \quad (27)$$

Minimize J Subject to:

$$\begin{cases} K^{\min} \le K_{\mathrm{PSS}} \le K^{\max} \\ T_1^{\min} \le T_1 \le T_1^{\max} \\ T_2^{\min} \le T_2 \le T_2^{\max} \\ T_3^{\min} \le T_3 \le T_3^{\max} \\ T_4^{\min} \le T_4 \le T_4^{\max} \end{cases} \quad (28)$$

where parameters N_g, N_P, t_{sim}, λ, ζ and β denote the number of generators, number of working conditions, the simulation time, the ith eigenvalue, the damping ratio for ith eigenvalue and balanced factor, respectively. Typical ranges for these parameters are [0.01–50] for $Kpss$ and [0.01–1] for T_1–T_4. The graphical flowchart of the proposed PSS design strategy is shown in Fig. 16. To evaluate the proposed PSS design, the various operating condition is considered as follows:

$$0.5 \le P_{g0} \le 1.5,\ 0.2 \le X_e \le 0.5. \quad (29)$$

Note that the eigenvalue must be to the left-hand side of the complex plane so as to guarantee the power system stability. According to BSSA design, the optimum value for PSS parameters is $K_{\mathrm{PSS}} = 30.35$, $T_1 = 0.0895$, $T_2 = 0.241$, $T_3 = 0.0687$ and $T_4 = 0.0013$. Figure 17 shows the dynamic responses for the generator. To make a fair comparison, CPSS (Shayeghi et al. 2011), PSS-based PSO (Shayeghi et al. 2011), PSS-based HBMO (Shayeghi et al. 2011) and PSS-based BSSA have been implemented in the same environment and tested under the same conditions.

As shown in Fig. 17, the superiority of the proposed designed stabilizer by BSSA is proved. As Eq (29), 44 operating conditions are examined, resulting, Fig. 18 presents the obtained disturbed eigenvalue by proposed stabilizers. When the reactance (X_e) is enlarged, eigenvalue tends to shift toward the source and increasing P forced the eigenvalue to shift the real part of the oscillatory mode to the right half complex plane.

According to the eigenvalue profile on the s-plane, it can be obvious that system without stabilizer is unstable. Negative damping values show the increasing oscillation when the corresponding mode is excited; therefore, it makes instability operation.

The eigenvalues with negative damping need further analysis before analyzing the rest of the system. The simulation results are consistent with the theoretical and simulation results and demonstrate that the proposed stabilizer outperforms the other stabilizers in the literature. It is quite obvious that the desired state of the SMIB system equipped with BSSA-PSS for all eigenvalues is in the left-hand side of the complex plane or s-plane.

To sum up, the presented results show that the speed deviation and the settling time of generator are much damped with employing the proposed BSSA-PSS stabilizer. In fact, its robustness and effectiveness are proved in damping transient oscillations.

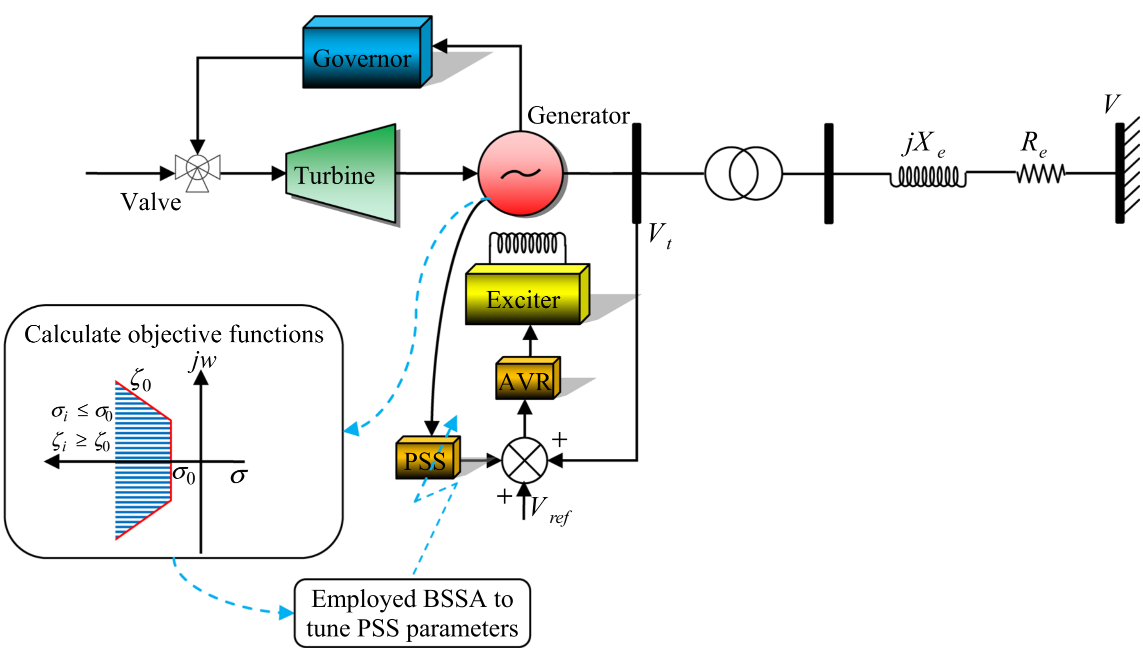

Fig. 16 Simplified graphical diagram of SMIB integrated the optimal design of PSS by BSSA

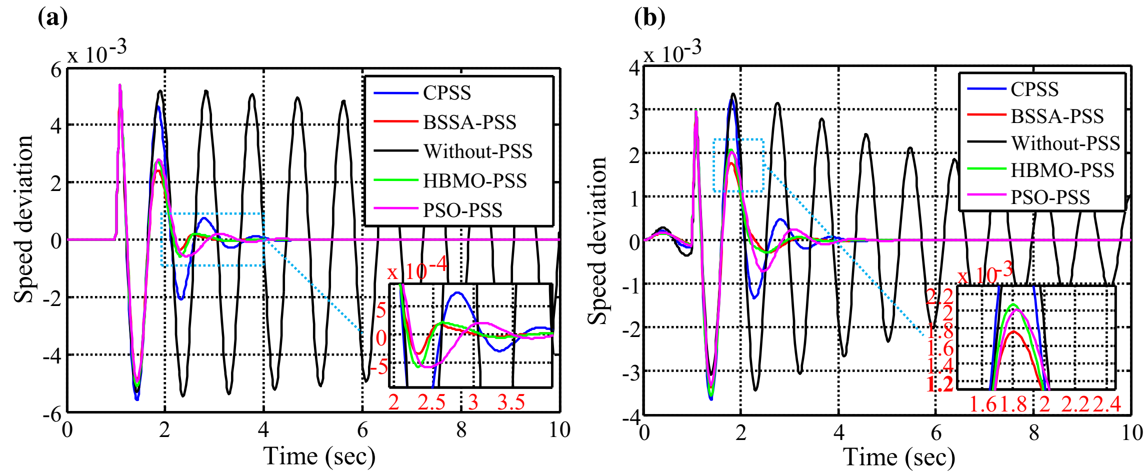

Fig. 17 Time-domain evaluation of the damping introduced by different types of PSS, **a** under active power 0.75 p.u. and three-phase fault in 1 s without open lines, **b** under active 0.5 p.u. and three-phase fault in 1 s, while line A is opened in this condition

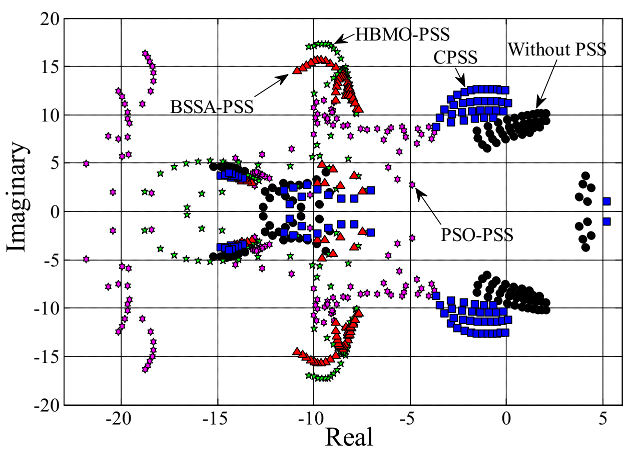

Fig. 18 Plot of eigenvalue for SMIB power system with different type stabilizers

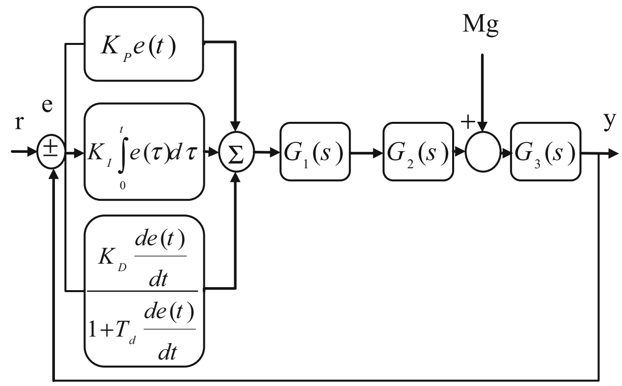

Fig. 19 The system structure of hydro-turbine governing system

3.2.2 Optimal design of hydro-turbine governing system

This system consists of the proportional–integral–derivative (PID), governing system, the BSSA algorithm, the hydraulic pressure servo system, the hydro-turbine system and the generator and load system that are shown in Fig. 19.

As shown in this figure, y and r denote the turbine rotating speed and input reference signals, respectively; $e = r - y$ is the error signal. The PID controller with low pass filter can be expressed by: in classical PID (CPID) controller equation is:

$$\mathrm{PID} = k_p + \frac{K_I}{S} + \frac{k_D S}{1 + T_d S}, T_d > > k_D. \tag{30}$$

The transfer function of the hydraulic pressure servo system is:

$$G_1(S) = \frac{1}{1 + T_y S}. \tag{31}$$

The hydro-turbine system can be expressed by:

$$G_2(S) = \frac{e_y - (e_{qy} e_y - e_{qh} e y) T_w S}{1 + e_{qh} T_w S}. \tag{32}$$

The generator and load can be modeled as follows:

$$G_3(S) = \frac{1}{T_a S + e_n} \tag{33}$$

where Mg is the disturbance or load. T_a and T_w are the inertial time constant and the current inertial time constant and other parameters are defined in Ghasemi et al. (2013). The transfer function of the PID governing system is

$$G_g(Z) = K_p + \frac{K_1}{1 - Z^{-1}} + K_D(1 - Z^{-1}). \tag{34}$$

Its incremental expression is

$$\Delta U(k) = K_p(e(k) - e(k-1)) + K_I e(k) + K_D(e(k) - 2e(k-1) + e(k-2)) \tag{35}$$

where $U(k)$ is the governor output and $e(k)$ is the frequency error of the kth sampling. The BSSA algorithm optimizes the three PID gains to improve the static and dynamic performances of the governor. The studied system based on optimal design of PID controller is shown in Fig. 20.

To show the ability of BSSA in this problem, the obtained results by BSSA are compared with some other optimization algorithms, such as CEP, FEP, MFEP and DCEMP (Chuanwen et al. 2006). The objective function is:

$$\begin{aligned}
&\text{Min}\, J = e^t e\\
&\text{subject to:}\\
&K_{P,\min} \le K_P \le K_{P,\max}\\
&K_{I,\min} \le K_I \le K_{I,\max}\\
&K_{D,\min} \le K_D \le K_{D,\max}
\end{aligned} \tag{36}$$

where $K_{P,\min}$, $K_{I,\min}$, $K_{D,\min}$ and $K_{P,\max}$, $K_{I,\max}$, $K_{D,\max}$ are the upper and lower bounds of K_P, K_I, K_D, respectively. The model of hydro-turbine is HL638-WJ-60, winding speed $n = 1000$ r/min, power $P_T = 1612$ kW, pipeline length $L = 1956$ m, cross-area 4.22 m^2, inertial time constant $T_a = 3.9$ s, current inertial time constant $T_w = 0.365$ s. The result simulation from BSSA, DCEMP, FEP, MFEP and CEP is presented in Table 6. According to the vane opening levels, the transfer coefficients are: $e_x = -0.728$, $e_y = 1.28$, $e_h = 0.95$, $e_{qx} = -0.075$, $e_{qy} = 0.956$, $e_{qh} = 0.618$. The test result of the a real hydro-turbine governing system shows that the BSSA can optimize the PID parameters efficiently, and the system has the characteristics of stability; low overshoot level and fast response. One of the main advantages of the BSSA is its robustness to the initial parameter settings. Also, the quality of the optimal solution does not related to the initial guess.

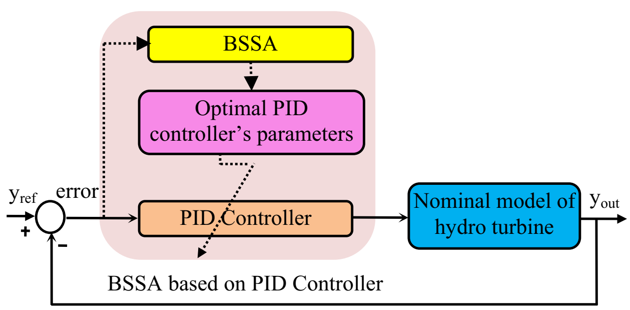

Fig. 20 The proposed PID controller design in hydro-turbine system

3.2.3 Economic load dispatch (ELD)

ELD is one of the basic engineering problems in the power system planning. The main purpose in this problem is optimal allocation of required power flows between available thermal units to minimize the total fuel cost while satisfy all the power units, load demand and diverse operating constraints (Ghasemi et al. 2016). The generator cost is formulated by quadratic functions, and the total fuel cost $F(P_G)$ in (\$/h) can be expressed as:

$$F(P_G) = \sum_{i=1}^{N} F_i(P_{G_i}) = \sum_{i=1}^{N} a_i + b_i P_{Gi} + c_i P_{Gi}^2 \tag{37}$$

where N is the number of generators; a_i,b_i and c_i are the cost coefficients of the ith generator; and P_{Gi} is the real power output of the ith generator. P_G is the vector of real power outputs of generators. The objective function corresponding to the production cost considering valve point loadings effects can be represented to be a more complex mathematical functions of the active power outputs from the generating units (Kohli and Arora 2018); one gets:

$$F_i(P_{G_i}) = a_i + b_i P_{G_i} + c_i P_{G_i}^2 + |e_i \sin(f_i(P_{G_i}^{\min} - P_{G_i}))| \tag{38}$$

Table 6 The result simulation from different types of optimization algorithms

Method	Vane opening (%)	Optimization parameters			T_S (s)	δ (%)	N
		K_P	K_I	K_D			
CEP (Chuanwen et al. 2006)	60	2.6	0.25	1.0	5.4	1.9	118
	80	4.0	0.23	0.2	6.0	1.77	97
FEP (Chuanwen et al. 2006)	60	2.7	0.23	1.5	5.2	1.85	101
	80	4.0	0.22	0.2	5.8	1.74	86
MFEP (Chuanwen et al. 2006)	60	3.0	0.25	1.8	4.8	1.35	57
	80	4.0	0.21	0.2	5.2	1.74	51
DCMEP (Chuanwen et al. 2006)	60	3.0	0.25	1.8	4.8	1.35	49
	80	4.0	0.21	0.2	5.2	1.74	30
BSSA	60	3.02	0.25	2.51	4.58	1.211	48
	80	3.72	0.21	1.24	4.98	1.603	55

Table 7 Comparison of operating cost ($/h) for 40 generators with different optimization algorithms

Algorithm	Operating cost ($/h)			SD	Time (s)
	Min	Mean	Max		
Two-phase neural network (Naresh et al. 2004)	105,236.0	NA	NA	NA	NA
BSA-EV (Dhillon et al. 2009)	102,355.4	NA	NA	NA	NA
PSO (NirbhowJap et al. 2016)	102,009.4	NA	NA	NA	4.123
PPO (NirbhowJap et al. 2016)	101,327.3	101,999.6	101,851.1	897.75	3.010
APSO (NirbhowJap et al. 2016)	101,295.8	102,057.2	103,851.1	987.34	3.257
SPPO (NirbhowJap et al. 2016)	101,143.7	11,710.6	103,851.1	69.77	2.891
AGWO (Singh and Dhillon 2019)	100,499.9	100,500.0	100,500.0	0.02037	111
Proposed BSSA	100,498.5	100,499.3	100,499.8	0.00198	2.76

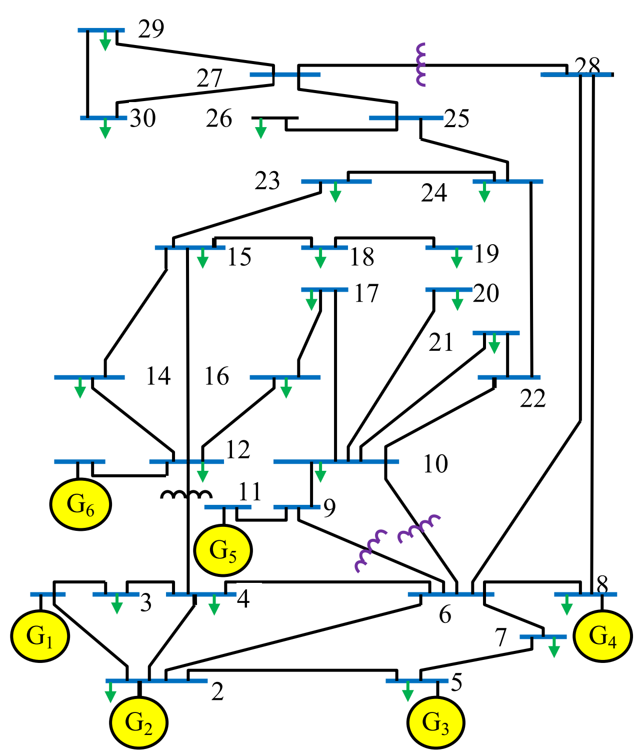

Fig. 21 Single line diagram of IEEE 30-bus system

where e_i and f_i are fuel cost coefficients for valve point effects. For stable operation, the real power output of each generator is restricted by lower and upper limits as follows:

$$P_i^{\min} \le P_i \le P_i^{\max}. \tag{39}$$

The total electric power generation must cover the total electric power demand P_D and the real power loss in transmission lines P_{loss}:

$$\sum_{i=1}^{N} P_i - P_D - P_{\text{loss}} = 0. \tag{40}$$

Calculation of P_{loss} implies solving the load flow problem (Naresh et al. 2004). In this engineering problem, consider 40 generators test system with power loss and valve point effects, while the obtained results by proposed BSSA algorithm comparing other methods from Naresh et al. (2004), Dhillon et al. (2009), NirbhowJap et al. (2016) and Singh and Dhillon (2019) are tabulated in Table 7. The simulation results show that the proposed method can get higher-quality solutions with short computational time.

3.2.4 Congestion management in restructured electricity market

In this section, the congestion management (CM) problem based on generator sensitivity and optimal rescheduling of active powers of generators is considered. Therefore, the values of generators are selected through generator sensitivity index. It is formulated as an optimization problem in deregulated electricity market and used BSSA to solve it at the minimum re-dispatch cost. The objective function can be expressed by:

$$\text{Min} \sum_{g=1}^{Ng} Ic_g(\Delta P_g).\Delta P_g \tag{41}$$

Subject to:

$$\begin{aligned} &\sum_{g=1}^{Ng} (GS_g^{ij}.\Delta P_g) + F_l^0 \le F_l^{\max} \\ &\Delta P_g^{\min} \le \Delta P_g \le \Delta P_g^{\max}; g = 1, 2, \ldots, Ng \\ &\Delta P_g^{\min} = P_g^{\min} - P_g; \Delta P_g^{\min} = P_g^{\min} - P_g \end{aligned} \tag{42}$$

where, IC_g is incremental/decremented cost of generator g. N_g and ΔP_g are number of the participating generators and active power adjustment at bus g, respectively. $\Delta P_{\min}^g$ and $\Delta P_{\max}^g$ denote the minimum and the maximum bounds of generator g. $P_{\min}^g$ and $P_{\max}^g$ are the minimum and the maximum generation limits of gth generator. Interested reader can refer to Abedinia et al. (2012) finding more details. To evaluate and compare BSSA performance, IEEE 30-bus system is used with six generators and forty one lines, while its single line diagram is shown in Fig. 21. In this test system, bus 1 is selected as the reference or slack bus. The congested lines are shown in Table 8.

Also, the values for all generators sensitivity are listed in Table 9. Since all generators have taken high constant values, all generators may as well consider for re-dispatch.

After optimization, the obtained average active powers are listed in Table 10.

According to 10 runs and obtained results of this table, the proposed BSSA is compared with CPSO (Abedinia et al. 2012), PSO-TVIW (Abedinia et al. 2012) and PSO-TVAC (Singh and Dhillon 2019) through some indices. BSSA finds the minimum re-dispatch cost solution of 205.32$/h, while PSO-TVAC, CPSO and PSO-TVIW provide 224.6952$/h, 237.9$/h, 240.3$/h and 239.2$/h, respectively. Also, the solutions of BSSA have the lowest standard deviation 1.26 (Zhou and Huang 2018).

Table 8 Congested line on the IEEE 30-bus system

Congested line	Active power flow (MW)	Line limit (MVA)	Overload (MW)
1 to 2	170	130	40

The studied cases were really challenging and needed a suitable balance between exploration and exploitation to find an accurate estimation of the global solution. In total, the obtained results of benchmarks and engineering problems prove the suitability of the BSSA algorithm. BSSA shows a proper convergence on many test functions. This is due to the position updating formulations and learning part based on the olfactory system. BSSA has the least control parameters compared to other optimization algorithms which may make a simple structure for the designer.

4 Conclusion

Since the modern engineering problems integrate the nonlinear and linear patterns, optimal operation with maximum efficiency converted an important issue for researchers, resulting, the nature-inspired method which imitates the specific behaviors of the nature phenomena that become a suitable way to avoid many mathematic formulations and save time. Hereby, this paper presents a novel nature-inspired meta-heuristic algorithm namely bear smell search algorithm which mimics the bear sense of smell strategy and the way bear moves to find food in thousand miles farther. This algorithm has two main parts, first, olfactory bulb mechanism that connected to the brain,

Table 9 Generator sensitivity for the IEEE 30-bus system

Gen no	1	3	5	8	11	13
GS 1-2	0	− 0.8908	− 0.8527	− 0.7394	− 0.7258	− 0.6869

Table 10 Comparison of BSSA solution with other methods for the IEEE 30-bus system

Algorithm	MW	$\Delta P1$	$\Delta P2$	$\Delta P5$	$\Delta P8$	$\Delta P11$	$\Delta P13$	Total ΔP	Cost ($/h)
CPSO (Abedinia et al. 2012)	Max	− 66.1	28.9	23.3	18.1	6.2	3.7	146.3	403.1
	Min	− 47.9	18.6	16.5	11.3	2.8	0.1	97.2	240.3
	Mean	− 55.9	22.6	16.2	10.5	5.6	2.6	113.2	287.1
	SD	8.3	7.6	3.5	3.3	3.2	3.3	15.9	48.2
PSO-TVIW (Abedinia et al. 2012)	Max	− 58.5	16.7	13.0	11.8	8.6	5.7	114.2	288.0
	Min	− 47.3	20.1	14.5	10.5	4.8	0.5	97.7	239.2
	Mean	− 50.1	18.9	13.2	9.2	5.9	4.1	101.4	253.1
	SD	2.8	3.5	5.4	3.3	3.5	6.1	13.3	3.8
PSO-TVAC (Abedinia et al. 2012)	Max	− 51.1	22.0	14.7	8.8	6.2	1.0	103.8	254.9
	Min	− 47.3	25.1	16.0	7.6	0.6	0.0	96.7	237.9
	Mean	− 49.3	17.5	14.0	9.9	6.8	3.0	100.5	247.5
	SD	0.8	2.1	2.1	2.2	2.3	2.4	4.6	1.6
BSSA	Max	− 45.91	20.65	7.17	2.77	0.17	16.03	92.70	226.86
	Min	− 43.61	20.31	16.00	6.72	0.63	0.01	87.28	205.32
	Mean	− 45.53	22.42	16.00	6.83	0.94	0.05	91.77	224.54
	SD	0.543	1.01	1.21	1.32	1.92	1.52	1.54	1.26

it analyze all received odors and followed the desired odor. This learning mechanism helps to find the best way at the least time, which means the fast convergence. The second part employed mesh mechanism to move the next position. These operators lead to an effective BBSA so as to find the global solution. The simulation and the obtained indices through numerical results by the constraint or unconstraint benchmark functions demonstrate that the proposed BSSA can generate solutions at an equal or outperforms other nature-inspired optimization algorithms; GA, PSO, GSA, ABC, HBMO, COA, LCA and SSA. Moreover, this outperformance is proved by a practical and real-world optimization problem based on the reported results in the figures and tables in this paper.

For the future work, the binary and multi-objective forms are expected to be presented for the combinational optimization in the real-world problems.

Albeit, the numerical and practical problems demonstrated that BSSA is a powerful population-based algorithm, but this is a fact that it is not the end; it is a new start point or idea to develop the BSSA or create the new optimization algorithms in order to tackle challenges in the real-world problems.

Compliance with ethical standards

Conflict of interest The author declares 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 author.

References

Abedinia O, Amjady N, Shayanfar HA, Ghasemi A (2012) Optimal congest management based VEPSO on electricity market. Int J Tech Phys Probl Eng (IJTPE) 4(2):56–62

Abedinia O, Amjady N, Ghasemi A (2016) A new metaheuristic algorithm based on shark smell optimization. Complexity 21(5):97–116

Alimoradi MR, Kashan AH (2018) A league championship algorithm equipped with network structure and backward Q-learning for extracting stock trading rules. Appl Soft Comput 68:478–493

Bansal JC, Gopal A, Nagar AK (2018) Stability analysis of artificial bee colony optimization algorithm. Swarm Evolut Comput 41:9–19

Cerdà V, Cerdà JL, Idris AM (2016) Optimization using the gradient and simplex methods. Talanta 148:641–648

Chuanwen J, Yuchao M, Chengmin W (2006) PID controller parameters optimization of hydro-turbine governing systems using deterministic-chaotic-mutation evolutionary programming (DCMEP). Energy Convers Manag 47(9–10):1222–1230

Dhillon JS, Dhillon JS, Kothari DP (2009) Economic-emission load dispatch using binary successive approximation-based evolutionary search. IET Gener Transm Distrib 3(1):1–16

dos Santos Júnior JG, do MonteLima JPS (2018) Particle swarm optimization for 3D object tracking in RGB-D images. Comput Graph 78:167–180

García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC'2005 Special Session on Real Parameter Optimization. J Heuristics 15:617

Ghasemi A (2013) A fuzzified multi objective interactive honey bee mating optimization for environmental economic power dispatch with valve point effect. Int J Electr Power Energy Syst 49:308–321

Ghasemi A, Shayeghi H, Alkhatib H (2013a) Robust design of multimachine power system stabilizers using fuzzy gravitational search algorithm. Int J Electr Power Energy Syst 51:190–200

Ghasemi A, Shayeghi H, Ahmadi R, Bazyar R (2013) VEPSO method with fuzzy-PID controller for nonlinear time-variant hydro-turbine governing systems. In: 5th electric power generation conference, pp 1–7

Ghasemi A, Gheydi M, Golkar MJ, Eslami M (2016a) Modeling of wind environment economic dispatch in power system and solving via an online learning meta-heuristic method. Appl Soft Comput 43:454–468

Ghasemi A, Shayeghi H, Moradzadeh M, Nooshyar M (2016b) A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management. Appl Energy 177:40–59

Grossman KJ, Mallik AK, Ross J, Kay LM, Issa NP (2008) Glomerular activation patterns and the perception of odor mixtures. Eur J Neurosci 27(10):2676–2685

Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Future Gener Comput Syst 101:646–667

Jain M, Singh V, Rani A (2019) A novel nature-inspired algorithm for optimization: squirrel search algorithm. Swarm Evolut Comput 44:148–175

Jordan R, Fukunaga I, Kollo M, Schaefer AT (2018) Active sampling state dynamically enhances olfactory bulb odor representation. Neuron 98(6):1214–1228

Kohli M, Arora S (2018) Chaotic grey wolf optimization algorithm for constrained optimization problems. J Comput Des Eng 5(4):458–472

Li Z (1990) A model of the olfactory bulb and beyond. California Institute of Technology Pasadena

Li Z, Hopfield JJ (1989) Modeling the olfactory bulb and its neural oscillatory processings. Biol Cybern 61(5):379–392

Li W, Cao J, Wu J, Huang C, Buyy R (2018) A collaborative filtering recommendation method based on discrete quantum-inspired shuffled frog leaping algorithms in social networks. Future Gener Comput Syst 88:262–270

Liang J, Qu B-Y, Suganthan PN (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Technical report, pp 1–32

Liu Q, Wu L, Xiao W, Wang F, Zhang L (2018) A novel hybrid bat algorithm for solving continuous optimization problems. Appl Soft Comput 73:67–82

Long W, Jiao J, Liang X, Tang M (2018) Inspired grey wolf optimizer for solving large-scale function optimization problems. Appl Math Model 60:112–126

Mareli M, Twala B (2018) An adaptive cuckoo search algorithm for optimisation. Appl Comput Inform 14(2):107–115

Mortazavi A, Toğan V, Nuhoğlu A (2018) Interactive search algorithm: a new hybrid metaheuristic optimization algorithm. Eng Appl Artif Intell 71:275–292

Naresh R, Dubey J, Sharma J (2004) Two-phase neural network based modelling framework of constrained economic load dispatch. IEEE Proc Gener Transm Distrib 151(3):373–378

NirbhowJap S, Dhillon JS, Kothari DP (2016) Synergic predator–prey optimization for economic thermal power dispatch problem. Appl Soft Comput 43(23):298–311

Patel VK, Savsani VJ (2015) Heat transfer search (HTS): a novel optimization algorithm. Inf Sci 324:217–246

Połap D, Woźniak M (2017) Polar bear optimization algorithm: meta-heuristic with fast population movement and dynamic birth and death mechanism. Symmetry 9:1–20

Rashedi E, Rashedi E, Nezamabadi-pour H (2018) A comprehensive survey on gravitational search algorithm. Swarm Evolut Comput 41:141–158

Shayeghi H, Ghasemi A (2012) Optimal design of power system stabilizer using improved ABC algorithm. Int J Tech Phys Probl Eng (IJTPE) 4(3):24–31

Shayeghi H, Ghasemi A (2014) A multi objective vector evaluated improved honey bee mating optimization for optimal and robust design of power system stabilizers. Int J Electr Power Energy Syst 62:630–645

Shayeghi H, Shayanfar HA, Akbarimajd A, Ghasemi A (2011) PSS design using an improved HBMO approach. In: 7th international conference on TPE (ICTPE-2011), Lefkosa, TR Northern Cyprus, pp 130–136

Shayeghi H, Shayanfar HA, Ghasemi A (2011b) A robust ABC based PSS design for a SMIB power system. Int J Tech Phys Probl Eng (IJTPE) 3(3):86–92

Shayeghi H, Ghasemi A, Moradzadeh M, Nooshyar M (2017) Day-ahead electricity price forecasting using WPT, GMI and modified LSSVM-based S-OLABC algorithm. Soft Comput 21(2):525–541

Singh D, Dhillon JS (2019) Ameliorated grey wolf optimization for economic load dispatch problem. Energy 169:398–419

Sirohi R, Singh A, Tarafdar A, Shahi NC (2018) Application of genetic algorithm in modelling and optimization of cellulase production. Biores Technol 270:751–754

Valipour K, Ghasemi A (2017) Using a new modified harmony search algorithm to solve multi-objective reactive power dispatch in deterministic and stochastic models. J Artif Intell Data Min 5(1):89–100

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82

Yamazaki K, Beauchamp GK, Singer A, Bard J, Boyse EA (1999) Odor types: their origin and composition. Proc Natl Acad Sci USA 96(4):1522–1525

Yang B, Yang S, Zhang J, Li D (2018) Optimizing random searches on three-dimensional lattices. Phys A 501:120–125

Zhang X, Kang Q, Cheng J, Wang X (2018) A novel hybrid algorithm based on Biogeography-Based Optimization and Grey Wolf Optimizer. Appl Soft Comput 67:197–214

Zhou X-P, Huang X-C (2018) Reliability analysis of slopes using UD-based response surface method combined with LASSO. Eng Geol 233:111–123

Zhou Y, Su K, Shao L (2018) A new chaotic hybrid cognitive optimization algorithm. Cogn Syst Res 52:537–542