Advances in Engineering Software 146 (2020) 102804

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems

Bikash Das[a,*], V. Mukherjee[b], Debapriya Das[c]

[a] Department of Electrical Engineering, Government College of Engineering and Textile Technology, Berhampore, West Bengal, India
[b] Department of Electrical Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India
[c] Department of Electrical Engineering, Indian Institute of Technology, Kharagpur, West Bengal, India

ARTICLE INFO

Keywords:
Benchmark function
CEC 2015
Global optimum solution
Optimization algorithm
Student psychology based optimization (SPBO)

ABSTRACT

In this article, a new metaheuristic optimization algorithm (named as, student psychology based optimization (SPBO)) is proposed. The proposed SPBO algorithm is based on the psychology of the students who are trying to give more effort to improve their performance in the examination up to the level for becoming the best student in the class. Performance of the proposed SPBO is analyzed while applying the algorithm to solve thirteen 50 dimensional benchmark functions as well as fifteen CEC 2015 benchmark problems. Results of the SPBO is compared to the performance of ten other state-of-the-art optimization algorithms such as particle swarm optimization, teaching learning based optimization, cuckoo search algorithm, symbiotic organism search, covariant matrix adaptation with evolution strategy, success-history based adaptive differential evolution, grey wolf optimization, butterfly optimization algorithm, poor and rich optimization algorithm, and barnacles mating optimizer. For fair analysis, performances of all these algorithms are analyzed based on the optimum results obtained as well as based on convergence mobility of the objective function. Pairwise and multiple comparisons are performed to analyze the statistical performance of the proposed method. From this study, it may be established that the proposed SPBO works very well in all the studied test cases and it is able to obtain an optimum solution with faster convergence mobility.

1. Introduction

Application of classical optimization methodology may not yield an optimum solution for complicated engineering problems. For that reason, in the last few decades, application of different metaheuristic optimization algorithms may be noticed to solve different complicated problems in various fields like science, engineering, management, economics etc. Metaheuristic optimization algorithms use an initial set of solutions to obtain the optimal solution. In the last few decades, lots of metaheuristic optimization algorithms have been proposed by the researchers. All of these algorithms use different methodologies to find out the optimum solution.

To solve various engineering problems, application of metaheuristic algorithms is very much common. One of the most common optimization algorithms is genetic algorithm (GA), proposed by Holland [1]. GA is based on the evolution theory of Darwin. Another very common nature inspired algorithm is particle swarm optimization (PSO), proposed by Kennedy and Eberhart [2]. PSO is inspired by the movement of swarm to find the best position within a closed boundary. Inspired by the natural behavior of honey bee to collect honey from the flowers, Kang et al. [3] proposed artificial bee colony algorithm. Ant colony optimization is based on foraging behavior of ant [4]. In the recent past, various nature inspired algorithms are available like tabu search [5, 6], harmony search (HS) [7, 8], bat algorithm [9], big bang-big crunch (BBBC) [10], plant growth simulation optimization [11], artificial immune system [12], bacterial foraging optimization (BFO) [13], shuffled frog-leaping algorithm (SFLA) [14], differential evolution (DE) [15], teaching learning based optimization (TLBO) [16, 17] and so on. HS has been proposed by Lee and Geem [7] to find an optimum solution for engineering problems with continuous designing variables. This algorithm is conceptualized using the musical process of search for the state of harmony. BBBC algorithm [10] is based on the theory of evaluation of the universe. On the other hand, BFO is inspired by the behavior of bacteria, called chemotaxis [13]. It is based on the local optimization performed by the bacteria where they try to climb up the nutrient concentration to avoid a noxious substance and search for a way out of neutral media. This algorithm is used to obtain global optimum solution but its performance is not up to the standard as compared to other

* Corresponding author.
E-mail addresses: bcazdas@gmail.com (B. Das), vivek_agamani@yahoo.com (V. Mukherjee), ddas@ee.iitkgp.ernet.in (D. Das).

https://doi.org/10.1016/j.advengsoft.2020.102804
Received 14 February 2020; Received in revised form 17 March 2020; Accepted 5 April 2020
Available online 12 May 2020

optimization algorithms used in the very recent days. It takes lots of time to execute. SFLA [14] is a population based optimization algorithm which performs local search and exchanges global information simultaneously. PSO is adopted to perform the local optimization search of SFLA. Based on the learning process of the students, Rao et al. [16, 17] proposed TLBO algorithm. This algorithm is divided into two phases named as, teacher phase where the learners use to gain knowledge from their teachers while the other phase is learner phase where the students use to gain knowledge by interacting with other students of the class. Yang and Deb [18, 19] have proposed cuckoo search (CS) optimization algorithm. It is inspired by natural behavior of the cuckoo bird and its process of laying eggs in the nest of some other birds.

Some other nature inspired optimization algorithms may also be noticed in the literature, such as grey wolf optimization (GWO) [20], animal migration optimization [21], dolphin echolocation algorithm (DEA) [22], virus colony search [23], flower pollination algorithm (FPA) [24], stochastic fractal search [25], gravitational search algorithm (GSA) [26]. GWO algorithm is inspired by the behavior of grey wolves and this algorithm is based on their leadership and hunting nature [20]. In GWO, the total population is divided into four types of wolves, named as alpha, beta, delta and omega. The hunting process is also divided into three main steps named as, searching for prey, encircling the prey and attacking the prey. The algorithm searches a global optimum solution based on the nature of the grey wolves. Based on the natural process of dolphin to locate the food using echo, Kaveh and Farhoudi [22] proposed DEA. FPA is also based on the natural process. It uses the natural process of flower pollination to identify optimum solution from the search space. Another important optimization algorithm is GSA [26] that works based on Newtonian gravity which states that *every particle in the universe attracts every other particle with a force that is directly proportional to the product of their mass and inversely proportional to the square of the distance between them*. Application of different hybrid optimization algorithms like GA-PSO [27], GA-fuzzy [28] etc. may also be found in different works to solve various kinds of optimization problems in the domain of science and engineering related applications.

Symbiotic organisms search (SOS) is a new optimization algorithm, proposed by Cheng and Prayogo in [29]. It is inspired by the relationship between different spices so as to survive and improve their fitness. SOS uses three different phases (*viz*., mutualism phase, commensalism phase and parasitism phase) to explain the relationship between various spices and their activity to survive (see [29]). Based on these three phases, SOS algorithm tries to obtain the global optimum solution for any problem. Another newly proposed nature inspired metaheuristic optimization algorithm is runner-root algorithm, proposed by Bayat in [30]. It is based on the function of runners and roots of some plants and their movement to search water and minerals from the soil. Based on the behavior of the grasshoppers, Saremi et al. have proposed grasshoppers optimization algorithm [31]. Many other optimization algorithms have been proposed in the recent days which include satin bower-bird optimizer [32], dragonflies algorithm [33], butterfly optimization algorithm (BOA) [34], social mimic optimization algorithm [35], poor and rich optimization (PRO) algorithm [36], black widow optimization algorithm [37], barnacles mating optimizer (BMO) [38] and so on.

Performance of most of the metaheuristic algorithms depends on their parameter selection [39]. Convergence properties of different algorithms are different. Some of the algorithms have faster convergence mobility of the objective function to obtain the global optimum solution. Convergence profile is measured against the number of iterations/number of fitness function evaluations (NFFEs). To find the optimum solution, it is desired that the optimization algorithms should take less NFFEs to obtain the optimum solution. At the same time, it should also take less computational time. But most of the available metaheuristic optimization algorithms exhibit slower convergence mobility and takes higher computational time to converge to the global optimum solution.

The main objective of this paper is to propose a new metaheuristic optimization algorithm which is based on the student psychology and an attempt has been made to explore the same to solve global optimization problems. Student psychology based optimization (SPBO) is inspired by the student psychology who is trying to obtain the highest marks/grade in the examination. Students always try to obtain good marks in the examination but the performance of a student depends upon her/his efficiency and interest to the subject offered to her/him. Based on this psychology, students try to give their best effort to improve their performance as well as try to obtain good marks aiming to be the best student in the class. To analyse the performance of the SPBO, the algorithm is applied to solve thirteen standard benchmark functions [23, 25, 26, 29]. Performance of the SPBO is compared to the performance of ten state-of-the-art optimization algorithms such as PSO, TLBO, CS, SOS, covariant matrix adaptation with evolution strategy (CMA-ES) [40], success-history based adaptive DE (SHADE) [41], GWO, BOA, PRO and BMO. The performance has been analysed based on the optimum result obtained as well as based on the convergence mobility it exhibits. Convergence profiles of the algorithms have been analysed in the present work as because the main objective of this paper is to present a new optimization algorithm that is capable to obtain the global optimum solution with faster convergence feature. Performance of the SPBO is further analysed by applying the algorithm to solve CEC 2015 problems [42] and the results are compared to the results obtained while using the aforementioned algorithms. The CEC-2015 problems are consisting of two unimodals, seven multimodals, three hybrid functions and three composition functions. Comparison study of all the adopted algorithms has been reported in this paper. Statistical comparison has been performed to demonstrate the performance of the SPBO in comparison to the performance exhibited by PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO.

The rest of the paper is organized as follows. The proposed SPBO algorithm is presented in the next section. Parameter selection and the importance of different phases of the SPBO are discussed in Section 3. Performance of the proposed SPBO to solve standard benchmark function as well as CEC 2015 problems are reported in Section 4. Section 5 presents a statistical analysis of the proposed SPBO algorithm. Finally, conclusions are drawn in Section 6. Some future directions are also included in this section.

2. SPBO: the proposed algorithm

Performance of a student is measured in terms of marks obtained in the examination. The student who obtained the highest marks in the examination is said to be the best student of the class and s/he is awarded accordingly for the same. Usually, students present in the class try to improve her/his performance for becoming the best student in the class. For that, the students need to give more effort to improve their performance in each subject offered to them. The proposed SPBO algorithm works on the psychology of the students who are trying to obtain the highest marks and try to be the best student by the way of improving their performance in the examination. The authors of the present work came to know about the stated psychology of the students after studying the behavior of the different students and talking to them. This study has been carried out in different schools, colleges as well as different universities of West Bengal, India for the last four years.

To be the best student in the class, students need to score more than the rest of the students present in the class. To accomplish this objective, they need to give more effort to the subjects offered to them. They need to perform well in each of the subjects so that their overall grade is improved. So, the students need to give effort subject wise to improve overall performance. But the effort given by any student to any subject depends on the students' capability, efficiency as well as interest in that subject. Therefore, it may be noted that improvement of performance in

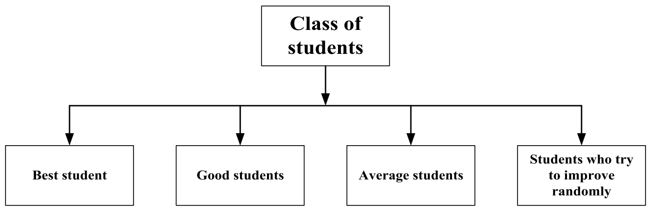

Fig. 1. Subject wise classification of the students.

the examination of all the students is not the same and may vary from student to student. To be the best student, the effort given by the students also depends on the students' psychology. Some of the students try to improve her/his performance by giving similar or better kind of effort given by the best student. At the same time, some of the students try to give efforts considering the effort given by the best student as well as try to give more efforts than the efforts given by the average students in the class. Improvement of the students' performance depends upon the effort given by them. As the amount of effort given by a student to a subject depends on the interest of the student to that subject, the students always will try to give more effort subject wise so that their overall grade in the examination is improved. Based on the above discussion, the students of a class may be subject wise categorized into four categories, as presented in Fig. 1.

A **Best student:** The student who obtained the highest overall marks/grade in the examination is said to be the best student in the class. The best student will always try to maintain her/his position by obtaining the highest marks in the class. To obtain the highest marks and to maintain her/his position, the best student needs to give more effort to each of the subjects than the efforts given by the rest of the students. So, it may be understood that the best student needs to give more effort than the effort given by any randomly selected student. Improvement of the best student may be expressed with the help of (1)

$$X_{bestnew} = X_{best} + (-1)^k \times rand \times (X_{best} - X_j) \tag{1}$$

where, X_{best} and X_j are, respectively, the marks obtained by the best student and randomly selected jth student in a particular subject, *rand* is a random number in between 0 and 1 and k is a parameter which is randomly selected as either 1 or 2. The main objective of the best student is to improve her/his performance in each subject so that her/his overall marks improve and s/he can maintain her/his best position.

B **Good student**: If a student finds interest in any subject, then s/he will try to give more and more effort to that subject so that her/his overall performance gets improved. This category of the student may be said, subject wise good student. The selection of this category of student is a random process because the psychologies of different students are different. To be the best student by obtaining the highest marks/grade in the examination, some of the students try to give similar or better kind of effort given by the best student. This category of the student may be represented with the help of (2a). At the same time, some of the students try to give more effort in their study than the effort given by the average students of the class as well as try to follow the effort given by the best student. This category of student may be expressed with the help of (2b)

$$X_{newi} = X_{best} + [rand \times (X_{best} - X_i)] \tag{2a}$$

$$X_{newi} = X_i + [rand \times (X_{best} - X_i)] + [rand \times (X_i - X_{mean})] \tag{2b}$$

where, X_i is the marks/grade obtained by the ith student in that subject, X_{mean} is the average performance of the class in that particular subject and *rand* is a random number in between 0 and 1.

C **Average student**: As the effort given by a student depends on the interest of the student in the subjects offered to them, the students will offer an average effort to that subject, if the student is less interested in some of the subjects. While giving an average effort to that subject, the students will try to give more effort to other subjects so that their overall marks get improved. This category of students may be said as the subject wise average student. Depending on different students' psychology, selection of this category of student is also a random process. Performance of this category of student may be represented by using (3)

$$X_{newi} = X_i + [rand \times (X_{mean} - X_i)] \tag{3}$$

where, X_i and *mean* are, in order, the marks obtained by the ith student and the average marks obtained by the class in that particular subject and *rand* is a random number in between 0 and 1.

D **Students who try to improve randomly**: Except these three aforementioned categories of students, some students try to improve their performance by themselves. They try to give effort to the subject randomly to some extent depending upon the subject. This category of student tries to give effort randomly to the subject so that overall performance in the examination improves. Performance of this category of the student may be expressed with the help of (4)

$$X_{newi} = X_{\min} + [rand \times (X_{\max} - X_{min})] \tag{4}$$

where, X_{min} and X_{max} are the minimum and the maximum limit of marks of the subject, respectively.

As described above, SPBO algorithm is based on the psychology of the student. Incorporating the aforesaid four types of psychologies of the students, the proposed SPBO may be represented with the help of the flowchart shown in Fig. 2. The pseudo-code of this algorithm is presented in Algorithm 1. As like other metaheuristics, SPBO works on an initial population which is analogues to the student of a class. Each population (student) is consisting of different variables which are analogues to the different subjects offered to them. Students try to give effort to the subjects in order to improve overall performance in the examination. The overall marks/grade obtained by a student may be said as the fitness of the population. The effort given by the students will be appreciated if the overall performance of the student in the examination improves. Similarly, change of variables will be accepted, if the fitness of the population improves. It has been considered in this algorithm that the students try to give their effort based on their interest to those subjects. It may happen that a student may find interest in some subject but s/he has less/average interest in some other subject. For example, let us consider a student is interested in a subject (say, X), so s/he will give more effort to the subject X. It may be said the student is a subject wise good student for the subject X. But on the other hand, it may happen that the considered student is to be average interested in another subject (say, Y). Similarly, the student may be less interested (subject wise) in some other subject (say, Z). The effort given by a student to a subject depends on the interest gained in the subject, which depends on the students' psychology and may vary from student to student and subject wise. It may be noted that subject wise interest gain is a random process. Based on this psychology, subject wise categorization of the students is considered as a random process in the proposed SPBO algorithm. Except the best student, subject wise selection of rest three categories of students (such as, good student, average student and students who try to improve randomly) has been considered as a random process. There is no numerical value to decide the structural decisions, which may be treated as an advantage of this algorithm. To ensure that the solutions lie within the feasible region, the modified population needs to be checked after undergoing modification at each stage to ensure whether the populations are within the feasible region or not. If any of the new population is found out to be out of the region, then that population needs to be bounded within the limits. In every stage, the performance of the class is calculated. The population having

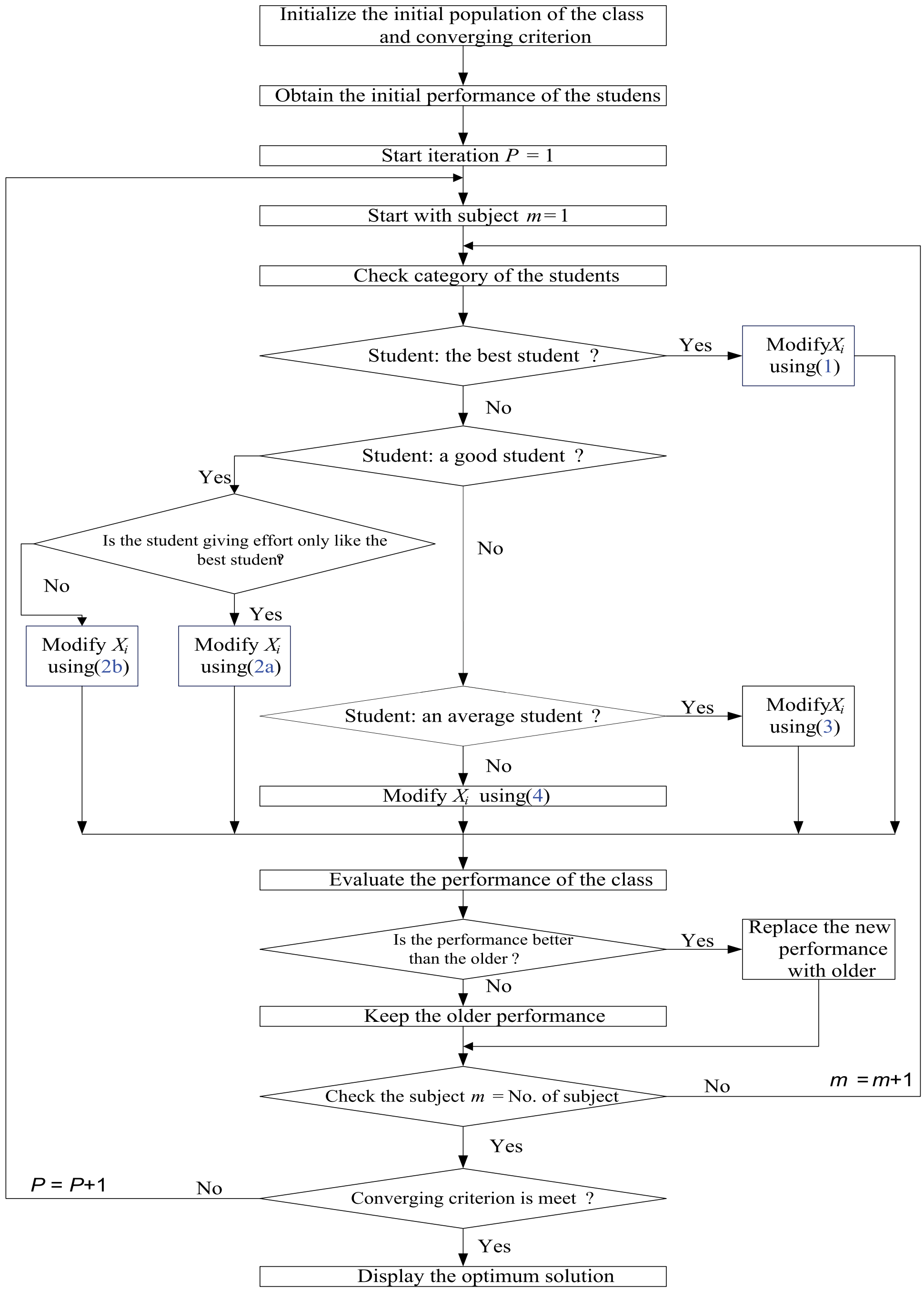

Fig. 2. Flowchart of the proposed SPBO algorithm.

the best fitness is considered as the best student in the class.

3. Parameter selection of SPBO and importance of its different phases

In the recent days, lots of optimization algorithms are available and are being used in different fields. But performances of most of the algorithms are dependent on the selection of certain parameters [39]. The parameters of the algorithms need to be adjusted properly to optimize the performance of the algorithm. Otherwise, it may not lead to the optimum solution. Tuning of the parameters is one of the major problems associated with the metaheuristic optimization algorithms available now-a-days.

However, the proposed SPBO algorithm does not have as such any tuneable parameter. The only thing that needs to be selected/adjusted for the SPBO is the population size. As such, selection of the population size is common for all the metaheuristic optimization algorithms. The population size needs to be adjusted according to the performance of the algorithms which varies with the number of unknown variables of the problem. The optimum population size also varies from algorithm to algorithm. To select optimum population size for the SPBO, the algorithm has been applied to solve five standard benchmark functions, namely Step, Sum square, Sphere, Rastrigin and Quartic.

The optimum population size has been determined using sensitivity analysis for all the aforementioned five benchmark functions. Sensitivity analysis has been done by varying the population size and analysing performance in terms of optimum solution achieved as well as based on the convergence mobility. For statistical analysis purpose,

```
Step 1 Initialize the population and converging criterion
Step 2 Evaluate the initial performance of the class
Step 3 while P <= Maximum iteration  do                          /* Iteration start*/
          for m = 1 to number of subject offered do
               i. for each student check the category of the student
                    if the student is the best student then
                         Modify performance using (1)
                    else
                         if the student is good student then
                              for each student check whether the student is
                                   following the best student only
                                   if Yes then
                                        Modify performance using (2a)
                                   else
                                        Modify performance using (2b)
                                   end if
                              end for
                         end if
                    else
                         if the student is the an average student then
                              Modify performance using (3)
                         end if
                    else
                         Modify performance using (4)
                    end if
               end for
               ii. Check the boundary
               iii. Evaluate the performance of the class
               iv. if the new performance is better then
                         Replace the old performance with the new one
                    else
                         Keep the old performance
                    end if
               m=m+1
          end for
          P=P+1
     end while
Step 4 Display the performance of the best student (optimum solution)
```

Algorithm 1. Pseudo-code of the proposed SPBO.

the algorithm has been run for twenty-five times for each set of the population for each benchmark functions considered here. For analysis purpose, it has been considered that the solution is converged when it reaches below 1E-5 and it has been also considered that a particular value of below 1E-5 is equal to zero. As because SPBO is, mainly, applied in this paper to optimize the solution for 30 dimension (30*D*) and 50*D* benchmark functions, so statistical sensitivity analysis has been done for both 30*D* as well as 50*D* benchmark functions to optimize population size.

Table 1 shows the statistical sensitivity analysis for 30*D* benchmark functions while Table 2 shows the same for 50*D* benchmark functions. From Tables 1 and 2, it may be noticed that the SPBO is able to offer an optimum solution in each run for all the five considered benchmark functions when the population size is considered above 15. But while considering convergence mobility, it may be noticed that the SPBO takes lesser NFFEs to converge when the population is considered to be 20 and 15 for 30*D* and 50*D* benchmark functions, respectively.

It is mentioned earlier that the proposed SPBO is based on the psychology of the students present in a class. The students present in the class have been classified based on their subject wise performance into four categories (*viz.*, best student, good student, average students and students who try to improve randomly). Similarly, the SPBO may be subdivided into four phases, named as best student, good student, average student and students who try to improve randomly. Performance of the SPBO depends on the performance of all the aforementioned four phases. A question may come into the readers' mind that how all these phases are important to build the algorithm. So, to analyse the importance and the contribution of each phase to the performance of the algorithm, the algorithm has been applied to all the five 50*D* benchmark functions considered in this study.

Fig. 3 shows the performance of the different phases of the SPBO while contributing to optimize performance of the algorithm for the considered five 50*D* benchmark functions. It may be noticed from this figure that the contribution of the best student to reach the optimum solution is much more for all the considered five benchmark functions. But it may be also observed that all the phases have their contribution to achieve an optimum solution and to reach the converging point. So, it may be apprehended that all the phases have their important roles in the performance of the algorithm. To shape the SPBO, all the four phases are required which helps the algorithm to obtain optimum solution with faster convergence mobility. If any of the phases does not work or is eliminated, then the algorithm may either not yield the desired optimal solution or it may take higher NFFEs to converge. Thus, the SPBO works well only when all the four phases are executed simultaneously.

4. Statistical analysis of SPBO for solving benchmark functions

In this section, performance of the SPBO has been analysed. The main objective of this research work is to propose an efficient algorithm which would be able to obtain global optimum solution with fast convergence mobility. To analyse the performance of the algorithm, it is applied to solve thirteen standard benchmark functions [23, 25, 26, 29] and fifteen benchmark functions of CEC 2015 [42]. The simulation is

Table 1
Statistical sensitivity analysis for 30*D* benchmark function.

Functions	Attributes	Population 10	15	20	25	30	35
Step	Best *FF*	0	0	0	0	0	0
	Worst *FF*	7.3487E-5	0	0	0	0	0
	Mean	2.5385E-5	0	0	0	0	0
	Lowest NFFEs to converge	17,110	16,665	15,620	16,525	17,130	17,885
	Highest NFFEs to converge	NA	17,565	16,820	18,025	18,030	18,935
	Average NFFEs to converge	NA	16,730	16,100	17,440	17,675	18,350
Sum square	Best *FF*	0	0	0	0	0	0
	Worst *FF*	5.6727E-4	0	0	0	0	0
	Mean	8.6482E-5	0	0	0	0	0
	Lowest NFFEs to converge	18,910	18,465	18,020	18,775	18,930	18,935
	Highest NFFEs to converge	NA	20,715	20,420	21,775	21,630	22,085
	Average NFFEs to converge	NA	20,145	19,760	20,385	20,535	20,742
Sphere	Best *FF*	0	0	0	0	0	0
	Worst *FF*	0	0	0	0	0	0
	Mean	0	0	0	0	0	0
	Lowest NFFEs to converge	22,210	22,065	21,020	24,775	25,230	25,235
	Highest NFFEs to converge	24,310	23,850	22,820	33,775	34,230	35,135
	Average NFFEs to converge	23,340	22,466	21,920	29,634	30,340	32,586
Rastrigin	Best *FF*	0	0	0	0	0	0
	Worst *FF*	1.6528E-5	0	0	0	0	0
	Mean	6.6309E-6	0	0	0	0	0
	Lowest NFFEs to converge	48,010	48,615	47,420	48,625	48,750	49,535
	Highest NFFEs to converge	NA	67,515	66,020	66,025	62,280	66,635
	Average NFFEs to converge	NA	59,030	57,980	58,765	59,120	59,815
Quartic	Best *FF*	0	0	0	0	0	0
	Worst *FF*	0	0	0	0	0	0
	Mean	0	0	0	0	0	0
	Lowest NFFEs to converge	8410	7665	7220	8275	9030	9485
	Highest NFFEs to converge	9910	8565	8420	9775	10,830	11,585
	Average NFFEs to converge	9340	8045	7880	8530	9520	9985

run on MATLAB 12b in a PC with Intel Core I5 (4th Gen) processor having a speed of 1.70 GHz with turbo boost up to 2.40 GHz and 8 GB of internal RAM. Twenty-five individual runs are performed for each of the benchmark functions and for each of the algorithms. The considered benchmark functions are 50*D* benchmark functions as described in Table 3. The considered benchmark functions are minimization

Table 2
Statistical sensitivity analysis for 50*D* benchmark function.

Functions	Attributes	Population 10	15	20	25	30	35
Step	Best *FF*	0	0	0	0	0	0
	Worst *FF*	4.9652E-4	0	0	0	0	0
	Mean	9.7903E-5	0	0	0	0	0
	Lowest NFFEs to converge	22,510	21,020	24,020	28,775	31,530	33,285
	Highest NFFEs to converge	NA	23,270	27,020	31,275	34,530	35,035
	Average NFFEs to converge	NA	22,295	25,705	29,775	32,890	33,810
Sum square	Best *FF*	0	0	0	0	0	0
	Worst *FF*	3.3461E-3	0	0	0	0	0
	Mean	5.4650E-4	0	0	0	0	0
	Lowest NFFEs to converge	29,500	27,770	30,020	32,525	34,530	36,750
	Highest NFFEs to converge	NA	29,270	34,020	35,025	39,030	40,280
	Average NFFEs to converge	NA	28,295	31,655	33,286	36,945	37,858
Sphere	Best *FF*	0	0	0	0	0	0
	Worst *FF*	5.2384E-5	0	0	0	0	0
	Mean	1.8737E-5	0	0	0	0	0
	Lowest NFFEs to converge	32,510	28,520	30,020	33,775	34,530	35,035
	Highest NFFEs to converge	NA	30,770	32,020	36,275	37,530	36,785
	Average NFFEs to converge	NA	29,645	30,235	34,867	35,620	36,130
Rastrigin	Best *FF*	0	0	0	0	0	0
	Worst *FF*	1.3496E-4	0	0	0	0	0
	Mean	6.8726E-5	0	0	0	0	0
	Lowest NFFEs to converge	64,010	63,010	66,020	68,750	64,530	68,285
	Highest NFFEs to converge	NA	93,010	91,020	96,275	94,530	96,285
	Average NFFEs to converge	NA	82,130	84,910	83,674	85,237	86,356
Quartic	Best *FF*	0	0	0	0	0	0
	Worst *FF*	0	0	0	0	0	0
	Mean	0	0	0	0	0	0
	Lowest NFFEs to converge	12,010	10,520	11,020	11,275	12,030	14,035
	Highest NFFEs to converge	14,510	12,770	14,020	13,775	13,530	15,785
	Average NFFEs to converge	13,840	12,020	12,560	13,050	13,360	14,450

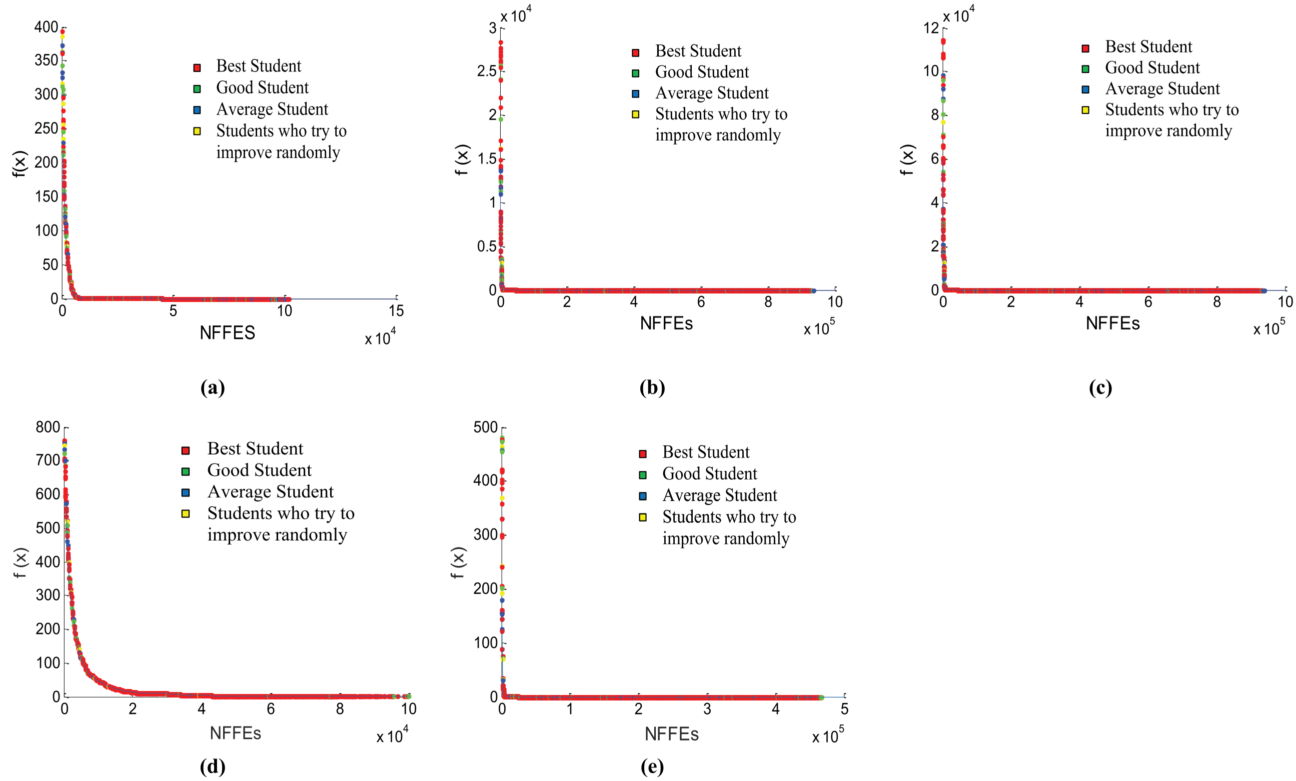

Fig. 3. Phase wise performance of the SPBO for benchmark functions: (a) Step, (b) Sum square, (c) Sphere, (d) Rastrigin and (e) Quartic.

Table 3
Details of the benchmark functions [23, 25, 26, 29].

Name of the function	Test function	Dimension (D)	Range	Optimum value	Type [37]
Step	$f_1(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$	50	[−5.12,5.12]	0	Unimodal
Sum square	$f_2(x) = \sum_{i=1}^{D} ix_i^2$	50	[−10, 10]	0	Unimodal
Sphere	$f_3(x) = \sum_{i=1}^{D} x_i^2$	50	[−100, 100]	0	Unimodal separable
Rastrigin	$f_4(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	50	[−5.12,5.12]	0	Multimodal separable
Quartic	$f_5(x) = \sum_{i=1}^{D} ix_i^4$	50	[−1.28,1.28]	0	Unimodal separable
Gricwank	$f_6(x) = \frac{1}{4000}[\sum_{i=1}^{D} x_i^2] - [\prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}})] + 1$	50	[−600, 600]	0	Multimodal non-separable
Schwefel 1.2	$f_7(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	50	[−100, 100]	0	Unimodal non-separable
Levy	$f_8 = \sum_{i=1}^{D-1} (x_i - 1)^2[1 + sin^2(3\pi x_{i+1})] + sin^2(3\pi x_1) + \lvert x_{n-1}\rvert[1 + sin^2(3\pi x_n)]$	50	[−10,10]	0	Multimodal non-separable
Non-continuous Rastrigin	$f_9(x) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10]$ where, $y_i = \left\{ \begin{array}{c} x_i \ if \ \lvert x_i\rvert < 1/2 \\ \frac{round\ (2x_i)}{2} \ if \ \lvert x_i\rvert > 1/2 \end{array} \right\}$	50	[−5.12,5.12]	0	Multimodal non-separable
Dixon Price	$f_{10} = (x_1 - 1)^2 + \sum_{i=2}^{D} i\,(2x_i^2 - x_{i-1})^2$	50	[−10,10]	0	Unimodal non-separable
Ackley	$f_{11}(x) = -20\exp(-0.2\sqrt{1/D\sum_{i=1}^{D} x_i^2}) - \exp(1/D\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	50	[−32, 32]	0	Multimodal non-separable
Schwefel 2.22	$f_{12}(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	50	[−10, 10]	0	Unimodal
Rosenbrock	$f_{13} = \sum_{i=1}^{D-1} [100\,(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	50	[−10,10]	0	Unimodal

function having the optimum value as 0. These benchmark functions include four unimodal types, two unimodal separable types, two unimodal non-separable types, two multimodal separable types and three multimodal non-separable types. Unimodal functions are those functions which have only one optimum solution. At that same time, the functions which include two or more local optima can be called as the multimodal functions. A function is separable if that function can be represented as a sum of functions of just one variable. Compared to the separable, getting the optimum solution of a non-separable function is more difficult as the accurate search depends on two or more associated variables. So, among the considered benchmark functions of Table 3, the multimodal non-separable functions are more difficult to get optimum solution as the search direction needs to be accurate for different variables as well as need to overcome the local optima to obtain the global optimum solution. The landscape representations of the considered benchmark functions with two variables have been portrayed in Fig. 4. Performance of the SPBO is compared to the results obtained by using PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO, and

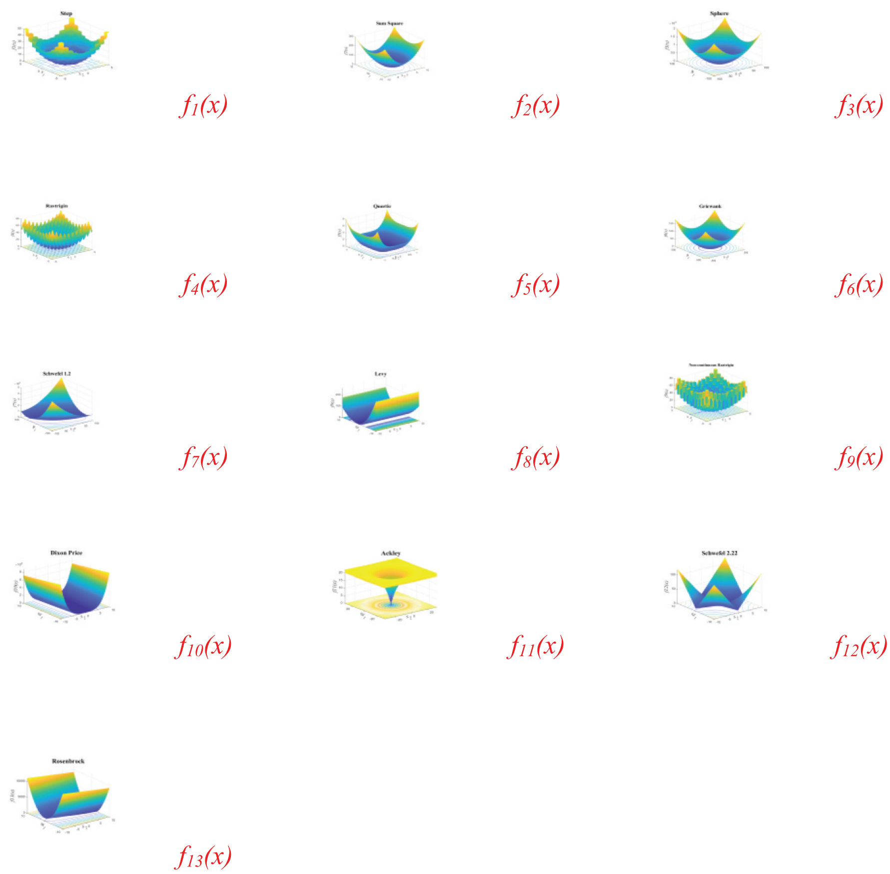

Fig. 4. Landscape of considered benchmark functions with two variables.

Table 4
Parameters of different algorithms for 50*D* benchmark functions problems.

Algorithms	Parameters	Population size	Maximum NFFE
PSO [2]	$C_1 = C_2 = 2$ $w_{max} = 0.9$, $w_{min} = 0.4$	100	10E+5
TLBO [16,17]	Teaching factor = either 1 or 2	100	10E+5
CS [18,19]	Levy co-efficient = 0.5 Discovery rate of alien eggs = 0.25	100	10E+5
SOS [29]	NA	100	10E+5
CMA-ES [40]	NA	160	10E+5
SHADE [41]	Crossover rate = 0.1	100	10E+5
GWO [20]	Component of coefficient vectors (a) = linearly decreasing from 2 to 0	100	10E+5
BOA [34]	Switch probability (p) = 0.8 Sensory modality (C) = 0.01 Power exponent (a) = 0.1	100	10E+5
PRO [36]	Mutation probability (Pmut) = 0.06	100	10E+5
BMO [38]	Percentage of characteristic of dad (p) = 0.6 Percentage of characteristic of mum (q) = 0.4 Penis length (Pl) = 60%	100	10E+5
SPBO [Proposed]	NA	15	10E+5

BMO. Parameters of the adopted eleven optimization algorithms are tabulated in Table 4.

Analysis of all the considered algorithms has been done based on the optimum results obtained as well as based on the convergence mobility. Convergence criterion is the same as stated in Section 3. Table 4 includes the parameter details of all the considered algorithms. It may be noticed that the population size is considered to be 100 for each of the algorithms like PSO, TLBO, CS, SOS, SHADE, GWO, BOA, PRO and BMO. For CMA-ES, the same is considered to be 160. But the SPBO does not require such large population size. The population of SPBO is considered as 15 only.

Statistical analysis of all the eleven optimization algorithms has been tabulated in Table 5. Performance ranking of all the algorithms has been done based on the optimum result obtained, as well as based on faster convergence mobility yielded. From Table 5, it may be focussed that performance of the SPBO is better as compared to the results obtained using PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO. Convergence mobility of the objective function for the SPBO is faster as compared to other optimization algorithms considered in this work. Out of thirteen 50*D* functions, the SPBO holds 1st rank for twelve problems and holds 2nd rank for f_9. The SOS holds 1st rank for f_9. So, it may be apprehended that the SPBO works better to find the optimum solution for higher dimensional problems as compared to other algorithms considered in this paper (*viz.*, PSO, TLBO, CS, SOS,

Table 5
Statistical analysis of SPBO for 50*D* benchmark functions test.

Functions	Attributes	PSO	TLBO	CS	SOS	CMA-ES	SHADE
$f_1(x)$	Best *FF*	6.1630E-33	1.4758E-12	2.5164E-18	0	0	0
	Worst *FF*	7.1419E-31	2.2205E-11	63.1688	1.8489E-32	0	0
	Mean	1.1213E-32	1.2807E-11	6.31688	8.0119E-33	0	0
	Std. Dev	2.0883E-31	6.3029E-29	18.95064	6.7793E-33	0	0
	Lowest NFFEs to converge	675,600	516,000	274,100	38,900	104,800	26,200
	Highest NFFEs to converge	698,400	555,200	NA	45,300	109,300	26,800
	Average NFFEs to converge	688,550	538,760	NA	41,660	107,300	26,600
	Rank	5	6	7	4	3	2
$f_2(x)$	Best *FF*	2.3337E-38	1.6636E-24	2.4948E-13	5.0113E-320	1.4246E-58	0
	Worst *FF*	7.4360E-35	7.1545E-23	27,000	1.2474E-316	5.6117E-56	5.0141E-320
	Mean	8.6204E-36	2.1255E-23	4390.0056	1.5599E-317	2.5300E-56	6.0105E-320
	Std. Dev	2.1949E-35	2.0774E-23	8903.9795	0	1.9906E-56	0
	Lowest NFFEs to converge	691,000	334,200	533,500	29,000	139,700	30,300
	Highest NFFEs to converge	739,400	366,800	NA	30,600	147,000	37,000
	Average NFFEs to converge	711,170	347,440	NA	29,560	142,940	34,780
	Rank	7	8	11	4	3	2
$f_3(x)$	Best *FF*	3.8917E-37	5.9581E-27	1.0123E-13	7.5389E-320	7.3454E-59	3.3010E-211
	Worst *FF*	5.4515E-35	2628.7	10,000	5.1096E-317	8.1619E-56	5.8916E-208
	Mean	9.6962E-36	262.87	2000.0000	1.0517E-317	2.9441E-56	2.5561E-208
	Std. Dev	1.6539E-35	788.61	3999.9999	0	3.3104E-56	0
	Lowest NFFEs to converge	710,200	278,200	520,300	29,800	140,000	39,800
	Highest NFFEs to converge	739,600	NA	NA	32,200	149,900	40,400
	Average NFFEs to converge	724,030	NA	NA	31,160	145,200	34,100
	Rank	7	10	11	3	6	4
$f_4(x)$	Best *FF*	46.7630	308.7908	125.3646	**0**	293.825	49,500
	Worst *FF*	82.5815	350.9752	189.0414	0	317.274	58,827
	Mean	63.4783	331.2454	155.0131	0	308.199	51,490.6
	Std. Dev	11.0148	11.9667	23.7936	0	3.9311	1641.78
	Lowest NFFEs to converge	NA	NA	NA	91,600	NA	NA
	Highest NFFEs to converge	NA	NA	NA	107,600	NA	NA
	Average NFFEs to converge	NA	NA	NA	96,080	NA	NA
	Rank	6	10	7	4	9	11
$f_5(x)$	Best *FF*	2.1222E-42	1.5395E-46	6.2503E-38	0	5.111E-116	0
	Worst *FF*	2.4198E-34	2.2063E-44	9.6397E-30	0	1.0488E-114	0
	Mean	2.5839E-35	4.6955E-45	9.7144E-31	0	5.1659E-115	0
	Std. Dev	7.2136E-35	7.0737E-45	2.8895E-30	0	3.5385E-115	0
	Lowest NFFEs to converge	732,600	152,000	145,100	11,400	52,320	25,600
	Highest NFFEs to converge	775,900	170,800	263,100	13,800	55,840	27,200
	Average NFFEs to converge	753,060	163,360	197,850	12,680	54,048	26,380
	Rank	8	7	9	2	6	4
$f_6(x)$	Best *FF*	0	1.0920E+3	0.1632	0	0	0
	Worst *FF*	0.0197	3.8014E+3	4.9636	0	0	0
	Mean	6.16E-3	2.3361E+3	1.5703	0	0	0
	Std. Dev	7.1554E-3	985.6775	1.6124	0	0	0
	Lowest NFFEs to converge	716,700	NA	NA	31,400	156,600	46,000
	Highest NFFEs to converge	NA	NA	NA	**33,000**	166,600	49,400
	Average NFFEs to converge	NA	NA	NA	**31,800**	161,560	47,680
	Rank	9	11	10	2	6	5

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
$f_1(x)$	Best *FF*	2.7500	7.2871	2.3744	9.3404	**0**
	Worst *FF*	3.3922	9.6943	2.7624	10.5241	**0**
	Mean	3.1364	8.9301	2.5525	10.0492	**0**
	Std. Dev	0.2328	0.6799	0.1103	0.3838	**0**
	Lowest NFFEs to converge	NA	NA	NA	NA	**21,020**
	Highest NFFEs to converge	NA	NA	NA	NA	**23,270**
	Average NFFEs to converge	NA	NA	NA	NA	**22,295**
	Rank	9	10	8	11	**1**
$f_2(x)$	Best *FF*	3.3801E-279	1.1335E-8	1.7611E-11	0	**0**
	Worst *FF*	1.5040E-272	1.2780E-8	9.9521E-10	0	**0**
	Mean	1.5617E-273	1.2109E-8	2.0935E-10	0	**0**
	Std. Dev	0	3.9265E-10	2.6869E-10	0	**0**
	Lowest NFFEs to converge	28,000	208,800	28,300	28,100	**27,770**
	Highest NFFEs to converge	30,100	213,400	47,300	31,400	**29,270**
	Average NFFEs to converge	28,870	210,940	37,000	29,670	**28,295**
	Rank	5	10	9	2	**1**
$f_3(x)$	Best *FF*	5.6813E-178	1.1587E-8	5.4504E-12	0	**0**
	Worst *FF*	1.3621E-172	1.4299E-8	2.5109E-11	0	**0**
	Mean	1.3994E-173	1.3080E-8	2.2501E-11	0	**0**
	Std. Dev	0	8.7324E-10	2.6568E-11	0	**0**
	Lowest NFFEs to converge	30,500	211,900	28,600	30,100	**28,520**
	Highest NFFEs to converge	40,100	219,400	32,600	32,100	**30,770**
	Average NFFEs to converge	34,980	215,970	30,240	31,320	**29,645**

(*continued on next page*)

Table 5 (*continued*)

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
	Rank	5	9	8	2	**1**
$f_4(x)$	Best *FF*	0	133.1806	2.3691E-9	**0**	**0**
	Worst *FF*	0	185.4647	4.2272E-9	0	**0**
	Mean	0	158.8897	3.3151E-9	0	**0**
	Std. Dev	0	15.6077	6.4375E-10	0	**0**
	Lowest NFFEs to converge	64,100	NA	68,500	65,100	**63,010**
	Highest NFFEs to converge	66,100	NA	98,100	79,100	**93,010**
	Average NFFEs to converge	65,100	NA	79,430	71,850	**82,130**
	Rank	2	8	5	3	**1**
$f_5(x)$	Best *FF*	1.2990E-292	7.9221E-11	1.7642E-7	0	**0**
	Worst *FF*	3.3763E-287	8.7668E-11	4.3861E-6	0	**0**
	Mean	9.2661E-288	8.3796E-11	1.1073E-6	0	**0**
	Std. Dev	0	2.7401E-12	1.1906E-6	0	**0**
	Lowest NFFEs to converge	19,700	85,900	10,800	12,100	**10,520**
	Highest NFFEs to converge	21,700	87,200	166,400	15,600	**12,770**
	Average NFFEs to converge	20,700	86,410	100,240	13,750	**12,020**
	Rank	5	10	11	3	**1**
$f_6(x)$	Best *FF*	0	1.4156E-8	2.5802E-13	0	**0**
	Worst *FF*	0	1.7296E-8	1.0378E-12	0	**0**
	Mean	0	1.5834E-8	6.3242E-13	0	**0**
	Std. Dev	0	9.6822E-10	2.2552E-13	0	**0**
	Lowest NFFEs to converge	39,300	229,500	34,100	32,100	**30,020**
	Highest NFFEs to converge	58,100	241,400	49,100	38,100	36,020
	Average NFFEs to converge	42,580	234,970	41,700	35,480	34,820
	Rank	4	8	7	3	**1**

Functions	Attributes	PSO	TLBO	CS	SOS	CMA-ES	SHADE
$f_7(x)$	Best *FF*	1.0142E-25	0.0065	0.0485	2.5040E-160	0.01959	3.2943E-133
	Worst *FF*	5.5683E-22	0.0274	191.7754	2.5997E-158	90.1684	1.0009E-130
	Mean	6.5959E-23	0.01672	56.9145	9.1233E-159	29.4824	4.9227E-131
	Std. Dev	1.6491E-22	6.3868×10^{-3}	65.7954	0	15.6264	4.1483E-131
	Lowest NFFEs to converge	738,500	NA	NA	57,800	NA	55,500
	Highest NFFEs to converge	764,800	NA	NA	60,200	NA	61,100
	Average NFFEs to converge	750,640	NA	NA	58,840	NA	58,360
	Rank	6	9	11	3	10	4
$f_8(x)$	Best *FF*	1.5963E-31	4.6392E-10	9.0500	1.3498E-31	0.75	0.8620
	Worst *FF*	0.1099	1.4162E-9	133.1021	0.5478	2.56	0.9659
	Mean	0.03297	8.9680E-10	29.5594	0.15156	1.226	0.8981
	Std. Dev	0.05036	3.1796E-10	44.1277	0.19567	0.3142	0.01884
	Lowest NFFEs to converge	696,500	684,800	571,500	50,900	NA	NA
	Highest NFFEs to converge	NA	659,800	NA	NA	NA	NA
	Average NFFEs to converge	NA	674,660	NA	NA	NA	NA
	Rank	3	2	7	4	5	6
$f_9(x)$	Best *FF*	40.0000	275.2162	156.0000	**0**	244.4171	0
	Worst *FF*	99.0002	333.9320	282.7500	**0**	285.006	0
	Mean	55.0055	309.5537	206.7750	**0**	259.4171	0
	Std. Dev	16.7332	18.2102	32.9504	**0**	6.0049	0
	Lowest NFFEs to converge	NA	NA	NA	**48,200**	NA	224,200
	Highest NFFEs to converge	NA	NA	NA	**63,400**	NA	229,200
	Average NFFEs to converge	NA	NA	NA	**54,200**	NA	226,920
	Rank	7	11	8	**1**	9	5
$f_{10}(x)$	Best *FF*	1.7341E-19	1.8725E-6	6.5711E-14	7.2107E-16	7.1523E-22	2.8174E-23
	Worst *FF*	9.6966E-19	3.7273	3.9193E+5	350.6326	8.1175E-19	5.5623E-21
	Mean	6.4502E-19	0.3764	6.0754E+4	70.9395	1.7723E-19	1.7736E-21
	Std. Dev	2.6596E-19	1.1170	1.1959E+5	135.0147	3.1796E-19	1.9498E-21
	Lowest NFFEs to converge	757,500	922,600	415,500	229,000	189,300	93,200
	Highest NFFEs to converge	811,000	NA	NA	NA	200,500	**190,400**
	Average NFFEs to converge	781,030	NA	NA	NA	194,660	164,300
	Rank	4	5	7	6	3	2
$f_{11}(x)$	Best *FF*	3.9968E-14	6.7972E-12	10.4431	8.8819E-16	2.2219E-14	2.6650E-14
	Worst *FF*	1/2168E-13	1.5622E-11	16.5022	4.4409E-15	1.5654E-13	9.5262E-8
	Mean	6.8890E-14	1.6969E-11	14.0138	3.3751E-15	6.6657E-14	3.6962E-8
	Std. Dev	2.1727E-14	6.8495E-12	1.7641	1.6281E-15	4.8007E-14	4.4103E-8
	Lowest NFFEs to converge	767,600	558,700	NA	85,800	199,800	366,400
	Highest NFFEs to converge	826,800	585,300	NA	120,200	204,600	397,600
	Average NFFEs to converge	803,150	572,740	NA	90,542	206,920	380,880
	Rank	6	8	10	4	5	7
$f_{12}(x)$	Best *FF*	1.0227E-26	4.8568E-8	9.7901E-5	4.9098E-163	4.7559E-28	5.3726E-214
	Worst *FF*	2.9559E-23	1.1355E-7	7.0478	1.1090E-160	9.8615E-25	1.8619E-210
	Mean	5.4492E-24	7.9764E-8	1.7258	2.4618E-161	2.2626E-25	5.4365E-211
	Std. Dev	9.7755E-24	2.2075E-8	2.2458	0	3.9123E-25	0
	Lowest NFFEs to converge	709,000	761,100	NA	45,000	236,300	56,100

(*continued on next page*)

Table 5 (*continued*)

Functions	Attributes	PSO	TLBO	CS	SOS	CMA-ES	SHADE
	Highest NFFEs to converge	762,600	808,500	NA	46,600	245,900	64,200
	Average NFFEs to converge	732,060	792,560	NA	45,560	240,100	69,900
	Rank	7	8	9	5	6	4

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
$f_7(x)$	Best *FF*	2.8667E-98	1.0799E-6	4.1709E-20	3.6398E-218	**1,3329E-229**
	Worst *FF*	1.7605E-96	1.1450E-6	1.2069E-5	5.6869E-214	**6.4152E-200**
	Mean	4.0219E-97	1.1195E-6	2.0234E-6	1.7625E-214	**6.4158E-201**
	Std. Dev	4.7881E-97	1.9646E-8	3.7152E-6	0	**0**
	Lowest NFFEs to converge	62,500	551,900	42,100	27,600	**18,020**
	Highest NFFEs to converge	64,900	561,600	NA	29,600	**24,020**
	Average NFFEs to converge	64,060	557,180	NA	28,510	**21,320**
	Rank	5	7	8	2	**1**
$f_8(x)$	Best *FF*	12.3304	31.9032	47.1226	48.6548	**1.3498E-31**
	Worst *FF*	23.2510	59.6850	49.8671	49.9301	**1.3498E-31**
	Mean	19.7011	46.7946	49.4788	49.2934	**1.3498E-31**
	Std. Dev	3.1301	10.0277	0.8241	0.3599	**0**
	Lowest NFFEs to converge	NA	NA	NA	NA	**26,270**
	Highest NFFEs to converge	NA	NA	NA	NA	**29,270**
	Average NFFEs to converge	NA	NA	NA	NA	**27,545**
	Rank	8	9	10	11	**1**
$f_9(x)$	Best *FF*	0	266.4056	4.9815E-10	0	0
	Worst *FF*	0	376.9597	1.5976E-6	0	0
	Mean	0	306.3152	1.6263E-7	0	0
	Std. Dev	0	29.5030	4.7833E-7	0	0
	Lowest NFFEs to converge	102,600	NA	104,600	12,900	102,000
	Highest NFFEs to converge	121,100	NA	133,600	16,900	125,000
	Average NFFEs to converge	114,950	NA	121,280	14,830	114,750
	Rank	9	8	11	10	2
$f_{10}(x)$	Best *FF*	427.9991	19.9730	887.3711	818.3997	**5.2340E-25**
	Worst *FF*	495.7177	34.1196	945.9448	1.0406E+3	**2.3628E-23**
	Mean	470.2174	25.6079	913.8215	920.2249	**4.7615E-21**
	Std. Dev	19.6364	3.4281	18.5837	62.7656	**6.7707E-24**
	Lowest NFFEs to converge	NA	NA	NA	NA	**66,020**
	Highest NFFEs to converge	NA	NA	NA	NA	259,500
	Average NFFEs to converge	NA	NA	NA	NA	**127,290**
	Rank	9	8	11	10	**1**
$f_{11}(x)$	Best *FF*	8.8818E-16	19.9113	1.5895E-7	8.8818E-16	**8.8818E-16**
	Worst *FF*	8.8818E-16	20.3097	7.2155E-7	8.8818E-16	**7.5495E-14**
	Mean	8.8818E-16	20.1816	5.6253E-7	8.8818E-16	**5.6161E-14**
	Std. Dev	0	0.1110	1.5850E-7	0	**2.8049E-24**
	Lowest NFFEs to converge	51,700	NA	48,100	45,900	**44,270**
	Highest NFFEs to converge	54,900	NA	150,100	47,400	**47,270**
	Average NFFEs to converge	52,860	NA	106,480	46,530	**45,470**
	Rank	3	11	9	2	**1**
$f_{12}(x)$	Best *FF*	4.3009E-225	5.6909E+65	6.6305E+15	1.8145E-220	**2.7145E-228**
	Worst *FF*	1.3447E-220	1.0647E+73	3.2227E+24	1.8706E-206	**1.8853E-220**
	Mean	7.3404E-220	1.1327E_72	3.5076E+23	1.8706E-207	**1.9603E-221**
	Std. Dev	0	3.1744E+72	9.5907E+23	0	**0**
	Lowest NFFEs to converge	14,500	NA	NA	23,200	**14,270**
	Highest NFFEs to converge	17,300	NA	NA	27,500	**22,520**
	Average NFFEs to converge	15,980	NA	NA	25,070	**19,745**
	Rank	2	11	10	3	**1**

Functions	Attributes	PSO	TLBO	CS	SOS	CMA-ES	SHADE
$f_{13}(x)$	Best *FF*	3.5885	4.5177	37.2911	24.9413	0.021241	0.0151
	Worst *FF*	100.8917	117.7085	38.2022	29.3954	9.4245	6.2736
	Mean	77.7876	52.7494	37.9221	26.8628	3.7628	2.2154
	Std. Dev	26.4423	36.8878	0.2796	1.6523	2.4615	2.4214
	Lowest NFFEs to converge	NA	NA	NA	NA	NA	NA
	Highest NFFEs to converge	NA	NA	NA	NA	NA	NA
	Average NFFEs to converge	NA	NA	NA	NA	NA	NA
	Rank	4	5	7	6	3	2

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
$f_{13}(x)$	Best *FF*	46.2375	48.7997	50.6342	48.7606	**0.0053**
	Worst *FF*	47.2070	48.9197	52.476	48.9374	**4.6280**
	Mean	46.5976	48.8672	48.8672	48.8606	**1.4891**
	Std. Dev	0.4315	0.0388	0.0388	0.0623	**1.9329**
	Lowest NFFEs to converge	NA	NA	NA	NA	NA
	Highest NFFEs to converge	NA	NA	NA	NA	NA
	Average NFFEs to converge	NA	NA	NA	NA	NA
	Rank	9	10	11	9	**1**

Table 6
CEC 2015 benchmark function [42].

Problem	Type	Function name	F(x*)	Bounds	Dimension (*D*)
F1	Unimodal	Rotated Bent Cigar function	100	$[-100,100]^D$	30
F2	Unimodal	Shifted Discus function	200	$[-100,100]^D$	30
F3	Simple multimodal	Shifted and Rotated Weierstrass function	300	$[-100,100]^D$	30
F4	Simple multimodal	Shifted and Rotated Schwefel's function	400	$[-100,100]^D$	30
F5	Simple multimodal	Shifted and Rotated Katsuura function	500	$[-100, 100]^D$	30
F6	Simple multimodal	Shifted and Rotated Happy Cat function	600	$[-100, 100]^D$	30
F7	Simple multimodal	Shifted and Rotated HGBat function	700	$[-100, 100]^D$	30
F8	Simple multimodal	Shifted and Rotated Expanded Griewank's plus Rosenbrock's function	800	$[-100, 100]^D$	30
F9	Simple multimodal	Shifted and Rotated Expanded Scaffer's function	900	$[-100, 100]^D$	30
F10	Hybrid function	Hybrid function 1 ($N = 3$)	1000	$[-100, 100]^D$	30
F11	Hybrid function	Hybrid function 2 ($N = 4$)	1100	$[-100, 100]^D$	30
F12	Hybrid function	Hybrid function 3 ($N = 5$)	1200	$[-100, 100]^D$	30
F13	Composition function	Composition function 1 ($N = 5$)	1300	$[-100, 100]^D$	30
F14	Composition function	Composition function 2 ($N = 3$)	1400	$[-100, 100]^D$	30
F15	Composition function	Composition function 3 ($N = 5$)	1500	$[-100, 100]^D$	30

CMA-ES, SHADE, GWO, BOA, PRO and BMO).

The study has been further carried out by applying the SPBO to solve CEC 2015 benchmark problems [42]. Chosen CEC 2015 benchmark problems are listed in Table 6. Among the listed fifteen CEC-2015 problems, there are two unimodal functions (F1, F2), seven are multimodal functions (F3-F9), three are hybrid functions (F10-F12) and three are composition functions (F13-F15). All these fifteen benchmark functions are of 30*D*. The parameters for this case are considered to be the same as those of the previous ones (refer Table 4), except the population size. The population size of PSO, TLBO, CS, SOS, CMA-ES and SHADE are considered as 50 for each, while the same is considered as 20 for the SPBO. To analyse the performance of all the eleven algorithms for CEC 2015 benchmark test, the main focus has been given on the optimum solution obtained.

The statistical results of all the fifteen benchmark functions for CEC 2015 have been reported in Table 7. From this statistical analysis, it may be noticed that the performance of the SPBO is better as compared to other considered algorithms. The SPBO is able to obtain minimum value for thirteen benchmark functions out of fifteen CEC 2015 benchmark functions whereas SOS and CS are able to obtain minimum value for two functions (F2 and F14) and one (F10) problem, respectively.

Performance of the SPBO is quite better for both the cases (*i.e.* 50*D* benchmark test function and CEC 2015 benchmark test function) as compared to other ten considered algorithms. The performance rank summery has been made in Table 8. It may be noticed from Table 8, that out of twenty-eight considered benchmark functions, the SPBO stands 1st rank for twenty-five benchmark functions and rank 2nd for two benchmark functions. From the study, it may be noticed that the SPBO is quite able to find the global optimum solution with faster convergence mobility. So, from this study, it may be set on that the performance of the SPBO is better in terms of optimum solution obtained as well as based on the convergence mobility.

5. Statistical analysis on performance of the proposed algorithm

To analyse the performance of the SPBO, statistical analysis has been carried out and presented in this section. This analysis has been done based on the performance of eleven considered optimization algorithms including the proposed one for the considered twenty-eight benchmark functions (thirteen 50D benchmark function, fifteen CEC 2015 benchmark functions). Two pairwise comparisons and two multiple comparisons have been carried out. For statistical analysis, it is considered that the performance of all the algorithms is equal to the null hypothesis.

5.1. Pairwise comparisons

The pairwise comparisons are performed based on direct comparisons of two algorithms when applied to a common set of problems. Two pairwise comparisons have been done. The first one is the sign test while the second one is the Wilcoxon sign test.

5.1.1. The sign test

This test is an easy and popular one to analyse the performance of an algorithm compared to some other algorithms. In this test, performances of two algorithms are compared based on performance exhibited by the algorithms for each case. Overall, the win of each algorithm is counted over the other algorithms. The algorithm which has higher overall win will be considered as better one as compared to the other [43]. Table 9 shows the statistical sign test performance comparison of the SPBO over the other ten algorithms considered in this study (*viz.*, PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO). It may be fascinating to note that the performance of the SPBO is better as compared to all other algorithms considered in this study. This result easily eliminates the null hypothesis. It may be also said that the performance of the SPBO is, significantly, better as compared to other ten considered optimization algorithms with a level of significance, $\alpha = 0.01$.

5.1.2. The Wilcoxon signed ranks test

The Wilcoxon signed ranks test is a nonparametric procedure which is used in hypothesis testing, involving design with two samples. This test is analogous to the statistical paired *t*-test. The Wilcoxon signed ranks test is applied to a pair to test results with an aim to detect a significant difference in performance between the two algorithms. This test is more sensitive than the *t*-test. It is based on commensurability of differences but only qualitatively: greater difference counts for more but the absolute magnitudes are ignored. The Wilcoxon signed ranks test does not assume normal distributions which is safer from the statistical point of view [43].

Table 10 shows the Wilcoxon signed rank test results for the performance of the SPBO as compared to PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO. It may be noted that $R+$ is the sum of the ranks for which the proposed algorithm outperforms the other algorithm while R- is the sum of ranks for the opposite. The $R+$, R- and *p*-values are to be calculated. An algorithm with $p < 0.05$ is considered as the significantly better algorithm as compared to the other one [43]. From Table 10, it may be perceived that the proposed algorithm outstripped the other algorithms considered in this study. It may be also noticed that performance of the SPBO is, significantly, better with $R+$ of 406 as compared to all the six algorithms (TLBO, CMA-ES, GWO, BOAm PRO and BMO) under consideration. $R+$ value in comparison to

Table 7
Statistical result of CEC-2015 benchmark functions.

Functions	Attributes	PSO	TLBO	CS	SOS	CMA-ES	SHADE
F1	Best *FF*	175.3717	7.1518E+3	106.3237	170.6122	254.7632	142.2494
	Worst *FF*	9.1755E+4	2.9472E+6	2.8305E+4	2.1051E+4	5.3257E+4	952.1983
	Mean	2.5810E+4	3.6295E+5	5.7590E+3	2.7799E+3	6.3212E+3	613.6201
	Std. Dev	2.7415E+4	8.6405E+5	8.6974E+3	6.1317E+3	9.3648E+3	104.9768
	Rank	5	7	2	4	6	3
F2	Best *FF*	3.2870E+3	9.9895E+4	4.7688E+3	**1.0404E+3**	6.9367E+3	3.8367E+3
	Worst *FF*	1.3921E+4	2.6509E+5	2.8877E+4	**3.2571E+3**	5.2645E+4	1.2024E+4
	Mean	7.4506E+3	1.5706E+5	1.4008E+4	**1.5845E+3**	1.8476E+4	7.0207E+3
	Std. Dev	3.5572E+3	4.6101E+4	7.3798E+3	**595.9260**	4.8729E+3	2.1424E+2
	Rank	3	10	5	**1**	6	4
F3	Best *FF*	332.7990	335.9985	324.5247	337.2679	329.8459	323.8341
	Worst *FF*	339.0979	339.7227	335.7574	340.4701	348.9271	349.1263
	Mean	336.4226	338.0126	330.6799	339.2599	335.9276	332.9347
	Std. Dev	1.9107	1.2813	3.2072	0.9139	3.9273	3.2583
	Rank	7	9	3	10	6	2
F4	Best *FF*	3.1021E+3	8.1955E+3	4.0036E+3	7.4996E+3	3.0186E+3	2.9385E+3
	Worst *FF*	7.4064E+3	8.4935E+3	6.5503E+3	8.3104E+3	7.1283E+3	3.7294E+3
	Mean	5.4050E+3	8.1325E+3	5.1252E+3	8.0005E+3	6.3628E+3	3.2918E+3
	Std. Dev	1.4811E+3	252.1199	672.5590	261.9115	1.3283E+3	616.9382
	Rank	4	11	5	8	3	2
F5	Best *FF*	500.1566	501.8146	500.4193	501.8815	500.4029	500.0101
	Worst *FF*	500.5564	502.6703	502.3282	502.8400	502.0928	501.8278
	Mean	500.3091	502.2348	501.1096	502.3780	501.3242	501.4692
	Std. Dev	0.1411	0.2572	0.5524	0.3170	0.5102	0.1482
	Rank	3	9	5	10	4	2
F6	Best *FF*	600.4862	602.9697	600.4700	600.3007	600.7301	600.2674
	Worst *FF*	603.9801	603.1151	600.8625	600.5803	601.0120	600.9347
	Mean	602.1054	603.0295	600.6457	600.4156	600.9346	600.6534
	Std. Dev	1.3189	0.0412	0.1139	0.0839	0.05729	0.5238
	Rank	6	8	5	3	7	2
F7	Best *FF*	727.3328	700.2432	700.2686	700.2474	700.3016	700.2471
	Worst *FF*	839.4139	700.8458	700.6071	700.7563	700.8617	700.5183
	Mean	760.2609	700.3669	700.3457	700.4746	700.6790	700.4792
	Std. Dev	34.0179	0.1672	0.0919	0.1796	0.1721	0.07612
	Rank	7	2	5	4	6	2
F8	Best *FF*	800.6886	800.6686	800.6435	800.6886	800.6886	800.1682
	Worst *FF*	801.0091	**800.6737**	801.0424	801.4536	801.2416	801.1371
	Mean	800.8458	**800.6886**	800.8289	801.0466	800.9618	800.9216
	Std. Dev	0.1012	0.01014	0.1486	0.2691	0.4792	0.03872
	Rank	5	4	3	7	6	2

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
F1	Best *FF*	5.9784E+9	6.2370E+10	4.2448E+10	7.4326E+10	**100.0660**
	Worst *FF*	1.9164E+10	9.9255E+10	7.0701E+10	1.3295E+11	**237.4924**
	Mean	1.1406E+10	7.8709E+10	5.9624E+10	1.1110E+10	**126.8694**
	Std. Dev	3.4638E+9	1.1245E+10	1.0600E+10	1.3295E+11	**38.5988**
	Rank	8	10	9	11	**1**
F2	Best *FF*	7.5278E+4	1.0748E+5	7.6785E+4	9.6708E+4	3.1058E+3
	Worst *FF*	2.2872E+5	1.7265E+6	1.5589E+5	2.8955E+5	8.8405E+3
	Mean	1.5020E+5	3.1572E+5	1..1517E+5	1.7349E+5	5.7838E+3
	Std. Dev	4.5746E+4	4.7515E+5	4.7515E+5	4.7766E+4	1.6483E+3
	Rank	7	11	8	9	2
F3	Best *FF*	326.9449	329.6540	335.8860	338.5750	**311.1241**
	Worst *FF*	337.7336	332.5694	343.2065	346.0173	**312.5560**
	Mean	333.0726	330.9097	340.5928	342.7962	**311.2769**
	Std. Dev	3.1317	0.7810	1.8827	2.1590	**1.2771**
	Rank	4	5	8	11	**1**
F4	Best *FF*	5.6444E+3	5.2141E+3	7.6136E+3	7.9078E+3	**2.2367E+3**
	Worst *FF*	7.2091E+3	6.7164E+3	8.7802E+3	8.9589E+3	**2.8415E+3**
	Mean	6.5640E+3	5.9051E+3	8.2928E+3	8.5185E+3	**2.6440E+3**
	Std. Dev	544.4922	505.7631	335.3417	314.2975	**179.5443**
	Rank	7	6	9	10	**1**
F5	Best *FF*	500.9960	501.1956	501.7966	502.1080	**500.0233**
	Worst *FF*	502.3101	501.9636	502.8002	503.5630	**500.1210**
	Mean	501.5160	501.7247	502.4272	502.7947	**500.1210**
	Std. Dev	0.3500	0.2118	0.2897	0.4165	**0.0247**
	Rank	6	7	8	11	**1**
F6	Best *FF*	600.4564	605.8027	604.3165	604.9072	**600.1899**
	Worst *FF*	602.1435	608.9812	606.9700	607.3757	**600.3326**
	Mean	600.7772	607.0063	606.0824	606.3753	**600.2959**
	Std. Dev	0.4686	0.9332	0.8983	0.6758	**0.04039**
	Rank	4	11	9	10	**1**
F7	Best *FF*	719.9466	895.7667	805.8922	896.0298	**700.2052**

(*continued on next page*)

Table 7 (*continued*)

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
	Worst *FF*	752.9300	1.1098E+3	1.0102E+3	1.0561E+3	**700.3518**
	Mean	732.3441	973.0312	914.5294	979.6075	**700.2962**
	Std. Dev	10.0397	60.3029	79.8543	45.6243	**0.03998**
	Rank	8	10	9	11	**1**
F8	Best *FF*	1.7816E+3	3.7544E+6	1.0590E+6	9.8228E+5	**800.0005**
	Worst *FF*	2.6640E+4	6.1971E+7	1.1063E+7	4.0011E+7	800.9847
	Mean	7.0066E+3	1.5606E+7	3.2875E+6	9.9080E+6	800.7080
	Std. Dev	7.2979E+3	1.7029E+7	2.7628E+6	1.14966E+7	0.3419
	Rank	8	11	10	9	1

Functions	Attributes	PSO	TLBO	CS	SOS	CMA-ES	SHADE
F9	Best *FF*	912.5777	913.3571	913.6469	911.8529	912.6281	910.8273
	Worst *FF*	914.0000	913.9503	914.4874	912.4558	915.9137	913.9277
	Mean	913.2622	913.6414	914.0375	912.1110	914.2768	912.2784
	Std. Dev	0.3966	0.2016	0.1927	0.1856	0.5738	0.03788
	Rank	5	9	10	3	6	2
F10	Best *FF*	4.8830E+3	2.6885E+6	**3.3801E+3**	1.9437E+4	6.7244E+4	7.7237E+3
	Worst *FF*	3.0468E+4	1.3930E+7	**1.4772E+4**	1.4024E+5	3.1426E+5	3.2761E+4
	Mean	1.1895E+4	8.8905E+6	**6.3040E+3**	5.9370E+4	2.6259E+5	2.0370E+4
	Std. Dev	6.9048E+3	3.4558E+6	**3.4906E+3**	4.0157E+4	4.6933E+4	7.1690E+3
	Rank	2	8	**1**	4	6	3
F11	Best *FF*	1.2251E+3	1.7597E+6	1.1462E+3	1.3213E+3	1.3477E+3	1.2003E+3
	Worst *FF*	4.8197E+3	1.1588E+7	1.4761E+3	1.9343E+3	1.5820E+3	1.5292E+3
	Mean	2.2782E+3	6.0297E+6	1.2357E+3	1.6718E+3	1.5002E+3	1.3918E+3
	Std. Dev	1.2549E+3	3.4036E+6	91.0205	207.4089	77.8297	96.9803
	Rank	4	8	2	5	6	3
F12	Best *FF*	1.3250E+3	2.3713E+3	1.4483E+3	1.2396E+3	1.4027E+3	1.2404E+3
	Worst *FF*	1.7122E+3	2.7666E+3	2.1880E+3	1.7579E+3	1.6284E+3	1.3920E+3
	Mean	1.3581E+3	2.5798E+3	1.9252E+3	1.4466E+3	1.5288E+3	1.3204E+3
	Std. Dev	459.7041	142.6849	228.3575	158.3214	80.0372	28.0394
	Rank	4	22	8	2	6	3
F13	Best *FF*	1.7285E+3	1.6620E+3	1.6864E+3	1.5655E+3	1.6729E+3	1.5652E+3
	Worst *FF*	2.3595E+3	1.7847E+3	1.6874E+3	1.6874E+3	1.7289E+3	1.6920E+3
	Mean	1.9599E+3	1.7248E+3	1.6429E+3	1.6429E+3	1.7027E+3	1.6828E+3
	Std. Dev	209.7510	41.9462	38.1671	38.1671	32.8930	102.8739
	Rank	8	5	7	3	6	2
F14	Best *FF*	1.5728E+3	1.6059E+3	1.5318E+3	**1.5000E+3**	1.5213E+3	1.5003E+3
	Worst *FF*	1.6397E+3	1.6397E+3	1.7604E+3	1.5738E+3	1.6002E+3	1.5892E+3
	Mean	1.6057E+3	1.6176E+3	1.6019E+3	1.5396E+3	1.5922E+3	1.5527E+3
	Std. Dev	20.8963	9.8301	59.7824	27.0004	66.9365	64.9387
	Rank	7	8	6	2	5	3
F15	Best *FF*	2.3545E+3	1.9402E+3	2.0139E+3	1.9020E+3	2.0032E+3	1.9307E+3
	Worst *FF*	2.6888E+3	2.5248E+3	3.1481E+3	2.2397E+3	2.4684E+3	1.9926E+3
	Mean	2.5638E+3	2.1433E+3	2.6991E+3	2.0476E+3	2.3020E+3	1.9552E+3
	Std. Dev	123.3976	221.4087	329.2993	124.1219	298.0387	46.9568
	Rank	9	6	8	2	7	4

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
F9	Best *FF*	912.3622	913.0358	913.8833	913.2866	**908.5773**
	Worst *FF*	913.7542	913.4165	913.4559	913.5350	**909.4733**
	Mean	912.8236	913.2829	913.1790	913.2866	**909.0722**
	Std. Dev	0.2618	0.1315	0.1987	0.1199	**0.2856**
	Rank	4	7	11	8	**1**
F10	Best *FF*	2.4633E+6	3.0879E+7	2.5108E+7	6.9101E+7	3.3826E+4
	Worst *FF*	3.2847E+7	4.6000E+8	2.3445E+8	7.4178E+8	1.7924E+5
	Mean	1.0390E+7	1.7600E+8	1.0636E+8	2.8775E+8	1.0284E+5
	Std. Dev	8.5776E+6	1.4302E+8	7.3591E+7	2.3208E+8	5.2370E+4
	Rank	7	10	9	11	5
F11	Best *FF*	1.3285 *E* + 3	4.8967E+8	3.7041E+7	1.5421E+8	**1.1376E+3**
	Worst *FF*	1.8203E+4	2.2316E+9	3.6642E+9	2.2850E+10	**1.2294E+3**
	Mean	4.6263E+4	1.2120E+9	1.1328E+9	4.3127E+9	**1.1923E+3**
	Std. Dev	6.1733E+4	5.9269E+8	1.3929E+9	6.5257E+9	**30.0745**
	Rank	7	10	9	11	**1**
F12	Best *FF*	1.3644E+3	1.4311E+3	1.5514E+3	1.9119E+3	**1.2247E+3**
	Worst *FF*	1.7927E+3	1.7628E+3	1.9199E+3	2.2799E+3	**1.2395E+3**
	Mean	1.4861E+3	1.5866E+3	1.7701E+3	2.0617E+3	**1.2335E+3**
	Std. Dev	131.3348	122.2909	103.9147	103.9147	**4.6940**
	Rank	54	7	9	10	**1**
F13	Best *FF*	1.6205E+3	2.2981E+3	2.0672E+3	2.1500E+3	**1.5631E+3**
	Worst *FF*	1.7635E+3	7.6169E+3	4.8209E+3	8.3803E+3	**1.6452E+3**
	Mean	1.6901E+3	3.6139E+3	2.9845E+3	3.8822E+3	**1.5760E+3**
	Std. Dev	41.8836	1.4251E+3	895.2883	1.9421E+3	**88.9730**
	Rank	4	11	9	10	**1**
F14	Best *FF*	1.5012E+3	1.7167E+3	1.6207E+3	1.6427E+3	**1.500E+3**

(*continued on next page*)

Table 7 (*continued*)

Functions	Attributes	GWO	BOA	PRO	BMO	SPBO
	Worst *FF*	1.5973E+3	2.2420E+3	1.7778E+3	2.0134E+3	**1.5451E+3**
	Mean	1.5312E+3	1.8948E+3	1.7027E+3	1.7526E+3	**1.5166E+3**
	Std. Dev	24.4292	167.4333	48.2480	125.1883	**16.3680**
	Rank	4	11	9	10	**1**
F15	Best *FF*	1.9317E+3	1.9278E+3	2.6789E+3	2.9625E+3	**1.9011E+3**
	Worst *FF*	2.0359E+3	3.0924E+3	3.0961E+3	3.3428E+3	**1.9068E+3**
	Mean	1.9923E+3	2.4031E+3	2.9572E+3	3.1496E+3	**1.9029E+3**
	Std. Dev	30.1802	388.9922	110.8066	128.6708	**1.6263**
	Rank	5	3	10	11	**1**

Table 8
Rank summary of the algorithms.

Functions	PSO	TLBO	CS	SOS	CMA-ES	SHADE	GWO	BOA	PRO	BMO	SPBO
$f_1(x)$	5	6	7	4	3	2	9	10	8	11	**1**
$f_2(x)$	7	8	11	4	6	3	5	10	9	2	**1**
$f_3(x)$	7	10	11	3	6	4	5	9	8	2	**1**
$f_4(x)$	6	10	7	4	9	11	2	8	5	3	**1**
$f_5(x)$	8	7	9	2	6	4	5	10	11	3	**1**
$f_6(x)$	9	11	10	2	6	5	4	8	7	3	**1**
$f_7(x)$	6	9	11	3	10	4	5	7	8	2	**1**
$f_8(x)$	3	2	7	4	5	6	8	9	10	11	**1**
$f_9(x)$	7	11	8	**1**	9	5	3	10	6	4	2
$f_{10}(x)$	4	5	7	8	3	2	9	8	11	10	**1**
$f_{11}(x)$	6	8	10	4	5	7	3	11	9	2	**1**
$f_{12}(x)$	7	8	9	5	6	4	2	11	10	3	**1**
$f_{13}(x)$	4	5	7	6	3	2	8	10	11	9	**1**
F1	5	7	2	4	6	3	8	10	9	11	**1**
F2	3	10	5	**1**	6	4	7	11	8	9	2
F3	7	9	3	10	6	2	4	5	8	11	**1**
F4	4	11	5	8	3	2	7	6	9	10	**1**
F5	3	9	5	10	4	2	6	7	8	11	**1**
F6	6	8	5	3	7	2	4	11	9	10	**1**
F7	7	2	5	4	6	3	8	10	9	11	**1**
F8	5	4	3	7	6	2	8	11	10	9	**1**
F9	5	9	10	3	6	2	4	7	11	8	**1**
F10	2	8	**1**	4	6	3	7	10	9	11	5
F11	4	8	2	5	6	3	7	10	9	11	**1**
F12	4	11	8	2	6	3	5	7	9	10	**1**
F13	8	5	7	3	6	2	4	11	9	10	**1**
F14	7	8	6	2	5	3	4	11	9	10	**1**
F15	9	6	8	2	7	4	5	3	10	11	**1**

Table 9
Statistical sign test comparison of the SPBO.

SPBO	PSO	TLBO	CS	SOS	CMA-ES	SHADE	GWO	BOA	PRO	BMO
Wins (+)	27	28	27	25	28	27	28	28	28	28
Loses (-)	1	0	1	3	0	1	0	0	0	0
Detected difference	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$	$\alpha = 0.01$

Table 10
The Wilcoxon signed rank test results.

SPBO	PSO	TLBO	CS	SOS	CMA-ES	SHADE	GWO	BOA	PRO	BMO
$R+$	399.5	406	398	392.5	406	391.5	406	406	406	406
R-	6.5	0	8	13.5	0	14,5	0	0	0	0
Level of significance	$p< 0.01$	$p< 0.01$	$p< 0.01$	$p< 0.01$	$p< 0.01$	$p< 0.01$	$p< 0.01$	$p< 0.01$	$p< 0.01$	$p< 0.01$

Table 11
The Friedman test result.

PSO	TLBO	CS	SOS	CMA-ES	SHADE	GWO	BOA	PRO	BMO	SPBO
5.64	7.68	6.75	4.21	5.82	3.54	5.57	8.96	8.89	7.79	1.21

PSO, CS, SOS and SHADE are 399.5, 398, 392.5 and 391.5, respectively. The p-values of the SPBO, as compared to the other algorithms, eliminates the null hypothesis and proves that the performance of the SPBO is, significantly, better with a significance level of 0.01.

5.2. Multiple comparisons

When performance of an algorithm needs to be analysed by comparing the achieved result with the results obtained using more than one algorithm, multiple comparisons are more preferable than the pairwise one. The multiple comparisons may be done based on the performance of the proposed algorithm as compared to all other considered algorithms. In this section, two multiple comparison tests have been done. The first one is multiple sign test while the second one is the Friedman test.

5.2.1. The Multiple sign test

This test allows identifying and highlighting the algorithm which is having statistically better performance compared to other algorithms under study [43]. From Table 9, it may be noticed that the performance of the SPBO is better as compared to all other algorithms considered in this study. From this test, it may be found that the SPBO wins most of the cases when compared to rest of the algorithms which proves that the performance of the SPBO is, significantly, better as compared to rest of the algorithms with a significance level of 0.01. This study also rejects the null hypothesis.

5.2.2. The Friedman test

This test is one of the popular tests to highlight better algorithm among a few considered algorithms. The Friedman test ranks the algorithms for each problem separately. In this test, the algorithms are ranked based on their performances. The best performance is considered as rank 1, the second best as rank 2 and so on. In case of ties, the average rank is considered. Thus, the overall average rank is calculated for each algorithm [43].

The Friedman test results have been tabulated in Table 11, which shows that the average rank of eleven algorithms based on their performances to solve the problems. Table 11 shows that the average rank of the SPBO is 1.21, which is only possible if it stands first in most of the cases. So, performance wise it may be said that the SPBO is, significantly, superior as compared to PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO. Better performance of the SPBO easily rejects the null hypothesis and proves its superiority over the other ten considered algorithms.

5.3. Inference from statistical analysis

For all the tests (*i.e.* pairwise comparisons and multiple comparisons), the SPBO proves pre-eminence in all the benchmark problems over all other optimization algorithms considered in this study. From this statistical comparison test, it may be inferred that the SPBO is better as compared to PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO counterparts.

6. Conclusion and scope of future work

In this work, a new nature inspired optimization algorithm is proposed. The proposed SPBO algorithm is based on the students' psychology to obtain the highest marks/grade in the examination. This algorithm does not have any adjustable parameter that may influence its performance. To analyse the performance of the proposed algorithm, it is applied to solve thirteen 50*D* standard benchmark functions. The obtained results using the SPBO are compared to the results obtained using ten other state-of-the-art optimization algorithms (*viz.*, PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO). For a fair comparison, the obtained results of all the eleven algorithms are analysed based on the optimum results obtained as well as based on the convergence mobility (*i.e.* NFFEs taken to converge). From this study, it is seen that the SPBO holds 1st rank for twelve benchmark functions out of thirteen considered 50D benchmark functions. It is found that the SPBO works better in terms of capability to obtain the optimum result as well as based on faster converging mobility as compared to other algorithms considered in this study. The study has been further extended by applying the SPBO to solve CEC 2015 benchmark functions. The obtained SPBO based results are compared to the results obtained using other ten considered benchmark functions. From the comparison study, it is noticed that the SPBO holds 1st rank for thirteen benchmark functions out of fifteen CEC 2015 benchmark functions. The performance of the SPBO is found to be better than the other algorithms considered in this work.

To analyse the performance of the SPBO, statistical tests are performed which includes two pairwise comparison tests (*i.e.* the sign test and the Wilcoxon signed ranks test) and two multiple comparisons tests (*i.e.* the multiple sign test and the Friedman rank test). For all the considered cases, performance of the SPBO is found to be better as compared to that of other ten considered optimization algorithms. It may be said from this study, that the proposed SPBO algorithm is better as compared to other considered algorithms with a significance level of 0.01.

From the benchmark function test (both 50*D* benchmark functions as well as CEC 2015), it may be concluded that the SPBO is able to obtain the optimum solution with faster convergence mobility and its performance is better as compared to PSO, TLBO, CS, SOS, CMA-ES, SHADE, GWO, BOA, PRO and BMO.

To sustain in the flood of optimization algorithm, a newly proposed algorithm needs to be novel as well as it should has better performance. The performance of the SPBO, as presented in this work, proves its superiority over the previous algorithms in terms of achievement of global optimum solution as well as faster convergence mobility. So, it may be concluded that the proposed SPBO algorithm may be applied by the future researchers in some other domains like engineering, science as well as management.

CRediT authorship contribution statement

Bikash Das: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Project administration. **V. Mukherjee:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing - original draft, Writing - review & editing, Supervision. **Debapriya Das:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing - original draft, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

[1] Holland H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology. Control and artificial intelligence. Oxford, England: University of Michigan Press; 1975.
[2] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE international conference on neural networks. 1995. p. 1942–8.
[3] Kang F, Li J, Li H. Artificial bee colony algorithm and pattern search hybridized for global optimization. Appl Soft Comput 2013;13(4):1781–91.
[4] Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1997;1(1):53–66.
[5] Glover F. Tabu search-part 1. ORSA J Comput 1989;1(3):190–206.
[6] Glover F. Tabu search-part 2. ORSA J Comput 1990;2(1):4–32.
[7] Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng

2005;194(36–38):3902–33.
[8] Das S, Mukhopadhyay A, Roy A, Abraham A, Panigrahi BK. Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization. IEEE Trans Syst Man Cybern – Part B: Cybern 2011;41(1):89–106.
[9[Yang XS. Bat algorithm for multi-objective optimization. Int . Bio-Inspired Comput 2011;3(5):267–74.
[10] Erol OK, Eksin I. A new optimization method: big bang-big crunch. Adv Eng Softw 2006;37(2):106–11.
[11] Rao RS, Narasimham SVI, Ramalingaeaju M. Optimal capacitor placement in a radial distribution system using plant growth simulation algorithm. Int. J. Electr. Power Energy Syst. 2011;33(5):1133–9.
[12] De Castro LN, Timmis J. Artificial immune systems: a new computational intelligence approach. Springer; 2002. ISBN 978-1-85233-594-6.
[13] Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 2002;22(3):52–67.
[14] Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Manag 2003;129:210–25.
[15] Price KV, Rainer M, Lampinen JA. Differential evolution: a practical approach to global optimization. Springer-Verlag; 2005.
[16] Rao RV, Savsani VJ, Vakharia DP. Teaching-learning based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 2011;43(3):303–15.
[17] Rao RV, Patel V. Comparative performance of an elitist teaching-learning based optimization algorithm for solving unconstrained optimization problems. Int J Ind Eng Comput 2013;4(1):29–50.
[18] Yang XS, Deb S. Cuckoo search via Lévy flights. Proceedings of the world congress nature & biologically inspired computing (NaBIC 2009). 2009. p. 210–4.
[19] Yang XS, Deb S. Engineering optimization by cuckoo search. Int J Math Model Numer Optim 2010;1(4):330–43.
[20] Mirjalili S, Mirjalili SM, Lewis A. Gray wolf optimizer. Adv Eng Softw 2014;69:46–61.
[21] Li X, Zhang J, Yin M. Animal migration optimization: an optimization algorithm inspired by animal migration behaviour. Neural Comput Appl 2014;24(7):1867–77.
[22] Kaveh A, Farhoudi N. A new optimization method: dolphin echolocation. Adv Eng Softw 2013;59:53–70.
[23] Li MD, Zhao H, Weng XW, Han T. A novel nature-inspired algorithm for optimization: virus colony search. Adv Eng Softw 2016;92:65–88.
[24] Yang XS. Flower pollination algorithm for global optimization. Proceedings of unconventional computation and natural computation. 7445. 2012. p. 240–9. Lecture Notes in Computer Science.
[25] Salimi H. Stochastic fractal search: a powerful metaheuristic algorithm. Knowl Based Syst 2015;75:1–18.
[26] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179:2232–48.
[27] Kao Y, Zahara E. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl Soft Comput 2008;8(2):849–57.
[28] Hsiao YT, Chen CH, Chien CC. Optimal capacitor placement in distribution systems using combination fuzzy-GA method. Int J Electr Power Energy Syst 2004;26(7):501–8.
[29] Cheng MY, Proyogo D. Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 2014;139:98–112.
[30] Merrikh-Bayat F. The runner-root algorithm: a metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. Appl Soft Comput 2015;33:291–303.
[31] Saremi S, Mirjalili S, Lewis A. Grasshopper optimization algorithm: theory and application. Adv Eng Softw 2017;105:30–47.
[32] Samarah Moosavi SH, Khatibi Bardsiri V. Satin bowerbird optimizer: a new optimization algorithm to optimiza ANFIS for software development effort estimation. Inf Sci 2017;60:1–15.
[33] Mirjalini S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective discrete nad multi-objective problems. Neural Comput Appl 2016;27:1053–73.
[34] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 2019;23:715–34.
[35] Balochian S, Baloochian H. Social mimic optimization algorithm and engineering application. Exp Syst Appl 2019;134:178–91.
[36] Samarah Moosavi SH, Khatibi Bardsiri V. Poor and rich optimization algorithm: a new human based and multi populations algorithm. Eng Appl Artif Intell 2019;86:165–81.
[37] Hayyolalam V, Pourhaji Kazem AA. Black Widow Optimization Algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Eng Appl Artif Intell 2020;87:103249.
[38] Sulaiman MH, Mustaffa Z, Saari MM, Daniyal H. Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. Eng Appl Artif Intell 2020;87:103330.
[39] Swarup KS. Genetic algorithm for optimal capacitor allocation in radial distribution systems. Proceedings of the 6th WSEAS international conference on evolutionary computing. 2005. p. 152–9.
[40] Reddy SS, Panigrahi BK, Kundu R, Mukherjee R, Debchoudhury S. Energy and spinning reserve scheduling for a wind-thermal power system using CMA-ES with mean learning technique. Int J Electr Power Energy Syst 2013;53:113–22.
[41] Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution. Proceedings of IEEE CEC. 2013. p. 71–8.
[42] Chen Q, Liu B, Zhang Q, Liang JJ, Suganthan PN, Qu BY. Problem definitions and evaluation criteria for CEC 2015 special session on bound constrained single -objective computationally expensive numerical optimization. Sendai, Japan: Sendai Int. Centre; 2015.
[43] Derrac J, Garcia S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 2011;1:3–18.