

# Social group optimization (SGO): a new population evolutionary optimization technique

Suresh Satapathy<sup>1</sup> · Anima Naik<sup>1</sup>

Received: 2 October 2015 / Accepted: 1 August 2016 / Published online: 23 August 2016  
© The Author(s) 2016. This article is published with open access at Springerlink.com

**Abstract** Social group optimization (SGO), a population-based optimization technique is proposed in this paper. It is inspired from the concept of social behavior of human toward solving a complex problem. The concept and the mathematical formulation of SGO algorithm is explained in this paper with a flowchart. To judge the effectiveness of SGO, extensive experiments have been conducted on number of different unconstrained benchmark functions as well as standard numerical benchmark functions taken from the *IEEE Congress on Evolutionary Computation 2005 competition*. Performance comparisons are made between state-of-the-art optimization techniques, like GA, PSO, DE, ABC and its variants, and the recently developed TLBO. The investigational outcomes show that the proposed social group optimization outperforms all the investigated optimization techniques in computational costs and also provides optimal solutions for most of the functions considered in our work. The proposed technique is found to be very simple and straightforward to implement as well. It is believed that SGO will supplement the group of effective and efficient optimization techniques in the population-based category and give researchers wide scope to choose this in their respective applications.

**Keywords** Optimization · Evolutionary computation · Benchmark functions · CEC 2005

## Introduction

Population-based optimization algorithms motivated from nature commonly locate near-optimal solution to optimization problems. Every population-based algorithm has the common characteristics of finding out global solution of the problem. A population begins with initial solutions and gradually moves toward a better solution area of search space based on the information of their fitness. Over the last few decades, numbers of successful population-based algorithms have been emerged for solving complex optimization problems. Some of the well-known population-based optimization techniques are comprehensively cited below, and readers can refer details in the respective papers. Genetic algorithms (GAs) [1], being the most popular ones, are based on genetic science and natural selection operators. The differential evolution (DE) [2] is based on similar concept of GA but it offers all solutions an equal chance irrespective of their fitness to get selected as parents, unlike GA, and has found to be recently very well known to optimization researchers. Bacteria foraging (BF) [3] based on the social foraging behavior of *Escherichia coli*, shuffled frog leaping (SFL) [4] inspired by natural memetics providing beauty of local search and global information exchange, simulated annealing (SA) [5] based on steel annealing process, and ant colony optimization (ACO) [6] motivated from the manners of real ant colony. A technique based on swarm behavior such as fish schooling and bird flocking in nature known as Particle Swarm Optimization (PSO) [7] has been widely researched and applied to various fields of engineering-allied subjects. Artificial bee colony (ABC) [8] algorithm based on the intelligent foraging behavior of honeybee swarm, the gravitational search algorithm (GSA) [9] based on the law of gravity and notions of mass interactions, cuckoo search [10] inspired by the obligate brood parasitism of

✉ Suresh Satapathy  
sureshsatapathy@ieee.org

<sup>1</sup> ANITS, Visakhapatnam, India

some cuckoo species by laying their eggs in the nests of other host birds (of other species) are gaining popularity as well among users. Biogeography-based optimization (BBO) [11] based on the idea of the migration strategy of animals or other species for solving optimization problems, the intelligent water drops (IWDs) [12] algorithm enthused from observing natural water drops that flow in rivers and find almost optimal paths to their destination, the firefly algorithm (FA) [13] inspired by the flashing behavior of fireflies in nature, the honey bee mating optimization (HBMO) [14, 15] algorithm inspired by the process of marriage in real honey bees, the bat algorithm (BA) [16] inspired by the echolocation behavior of bats are few more population-based techniques in this category. The harmony search (HS) [17] optimization algorithm inspired by the improvising process of composing a piece of music, the big bang–big crunch (BB-BC) optimization [18] based on one of the theories of the evolution of the universe [18], black hole (BH) [19] optimization inspired by the black hole phenomenon have also been extensively tried successfully for solving various problems in engineering. Recently, teaching–learning-based optimization (TLBO) [20] based on the effect of the influence of a teacher on the output of learners in a class is being extensively studied by researchers to solve a variety of optimization problems in engineering applications. Even though all these algorithms are good enough for solving optimization problems, however, issues like finding optimal solutions, providing fast convergence without over fitting (computational efforts), choosing and controlling algorithm parameters, algorithm stability and robustness, consistency in providing solutions, adaptability to wide variety of applications, etc., have been the subjects of extensive research in optimization community. To address the aforementioned issues, researchers have developed many variants of the above-mentioned optimization algorithms, and even hybridization of several algorithms has also been attempted.

In an attempt to address few challenges like computational efforts, optimal solutions and consistency in providing optimal solutions, this paper proposes a new optimization technique named social group optimization (SGO) based on the human behavior of learning and solving complex problems.

In this work, we have done extensive study to further investigate the performance of our proposed SGO algorithm on many simple benchmark functions as well as benchmark functions from CEC 2005 competitions. Many advanced versions of state-of-the art algorithms like PSO, DE and ABC etc., and their variants are simulated to compare the performances with SGO. Also, the performance of SGO is compared with recently developed TLBO algorithms. Convergence characteristics of SGO are presented in plots. Results are reported in Tables with the mean and standard deviation values for each algorithm on each function over several simulation runs. To compare the significance of the

proposed algorithm, we have done Wilcoxon's rank-sum statistical tests.

The remaining of the paper is organized as follows: in "Social Group Optimization", we give a comprehensive description of SGO algorithm. The next section "Implementation of SGO for optimization" discusses the implementation of SGO for optimization followed by discussion about "Experimental results". The paper concludes with further research in "Conclusion".

## Social group optimization (SGO)

There are many behavioral traits such as honesty, dishonesty, caring, compassion, courage, fear, justness, fairness, tolerance or respectfulness etc., lying dormant in human beings, which need to be harnessed and channelized in the appropriate direction to enable him/her to solve complex tasks in life. Few individuals might have required level of all these behavioral traits to be capable of solving, effectively and efficiently, complex problems in life. But very often, complex problems can be solved with the influence of traits from one person to other or from one group to other groups in the society. It has been observed that human beings are great imitators or followers in solving any task. Group solving capability has emerged to be more effective than individual capability in exploiting and exploring different traits of each individual in the group to solve a given problem. Based upon this concept, a new optimization technique is proposed which is named as social group optimization (SGO).

In SGO, each person (a candidate solution) is empowered with some sort of knowledge having a level of capacity for solving a problem. SGO is another population-based algorithm similar to other algorithms discussed in the previous section. For SGO, the population is considered as a group of persons (candidate solutions). Each person acquires knowledge and, thereby, possesses some level of capacity for solving a problem. This is corresponding to the 'fitness'. The best person is the best solution. The best person tries to propagate knowledge amongst all persons, which will, in turn, improve the knowledge level of the entire members in the group.

The procedure of SGO is divided into two parts. The first part consists of the 'improving phase'; the second part consists of the 'acquiring phase'. In 'improving phase,' the knowledge level of each person in the group is enhanced with the influence of the best person in the group. The best person in the group is the one having the highest level of knowledge and capacity to solve the problem. And in the 'acquiring phase,' each person enhances his/her knowledge with the mutual interaction with another person in the group and the best person in the group at that point in time. The

basic mathematical interpretation of this concept is presented below.

Let  $X_j, j = 1, 2, 3, \dots, N$  be the persons of social group, i.e., social group contains  $N$  persons and each person  $X_j$  is defined by  $X_j = (x_{j1}, x_{j2}, x_{j3}, \dots, x_{jD})$ , where  $D$  is the number of traits assigned to a person which determines the dimensions of a person and  $f_j, j = 1, 2, \dots, N$  are their corresponding fitness values, respectively.

### Improving phase

The best person (gbest) in each social group tries to propagate knowledge among all persons, which will, in turn, help others to improve their knowledge in the group.

Hence,  $\text{gbest}_g = \min\{f_i, i = 1, 2, \dots, N\}$  at generation  $g$  for solving **minimization problem**. (1)

In the improving phase, each person gets knowledge (here knowledge refers to change of traits with the influence of the best person's traits) from the group's best (gbest) person. The updating of each person can be computed as follows:

```

For i = 1 : N
  For j=1:D
     $X_{newij} = c * X_{oldij} + r * (\text{gbest}(j) - X_{oldij})$ 
  End for
End for
where  $r$  is a random number,  $r \sim U(0, 1)$ 
Accept  $X_{new}$  if it gives a better fitness than  $X_{old}$ .
  
```

(2)

where  $c$  is known as self-introspection parameter. Its value can be set from  $0 < c < 1$ .

### Acquiring phase

In the acquiring phase, a person of social group interacts with the best person (gbest) of that group and also interacts randomly with other persons of the group for acquiring knowledge. A person acquires new knowledge if the other person has more knowledge than him or her. The best knowledgeable person (here known as person having 'gbest') has the greatest influence on others to learn from him/her. A person will also acquire something new from other persons if they have more knowledge than him or her in the group.

The acquiring phase is expressed as given below: (3)

$$\text{gbest} = \min\{f(X_i), i = 1, 2, \dots, N\}$$

( $X_i$ 's are updated values at the end of the improving phase)

For  $i = 1 : N$

Randomly select one person  $X_r$ , where  $i \neq r$

If  $f(X_i) < f(X_r)$

For  $j = 1 : D$

$$X_{newi,j} = X_{oldi,j} + r_1 * (X_{i,j} - X_{r,j}) + r_2 * (\text{gbest}_j - X_{i,j})$$

End for

Else

For  $j = 1 : D$

$$X_{newi,:} = X_{oldi,:} + r_1 * (X_{r,:} - X_{i,:}) + r_2 * (\text{gbest}_j - X_{i,j})$$

End for

End If

Accept  $X_{new}$  if it gives a better fitness function value.

End for

(4)

where  $r_1$  and  $r_2$  are two independent random sequences,  $r_1 \sim U(0, 1)$  and  $r_2 \sim U(0, 1)$ . These sequences are used to affect the stochastic nature of the algorithm as shown above in Eq. (4).

For further clarity and ease of implementation, the entire process is now presented in an easy-to-understand flowchart (Fig. 1)

### Implementation of SGO for optimization

The step-wise procedure for the implementation of SGO is given in this section.

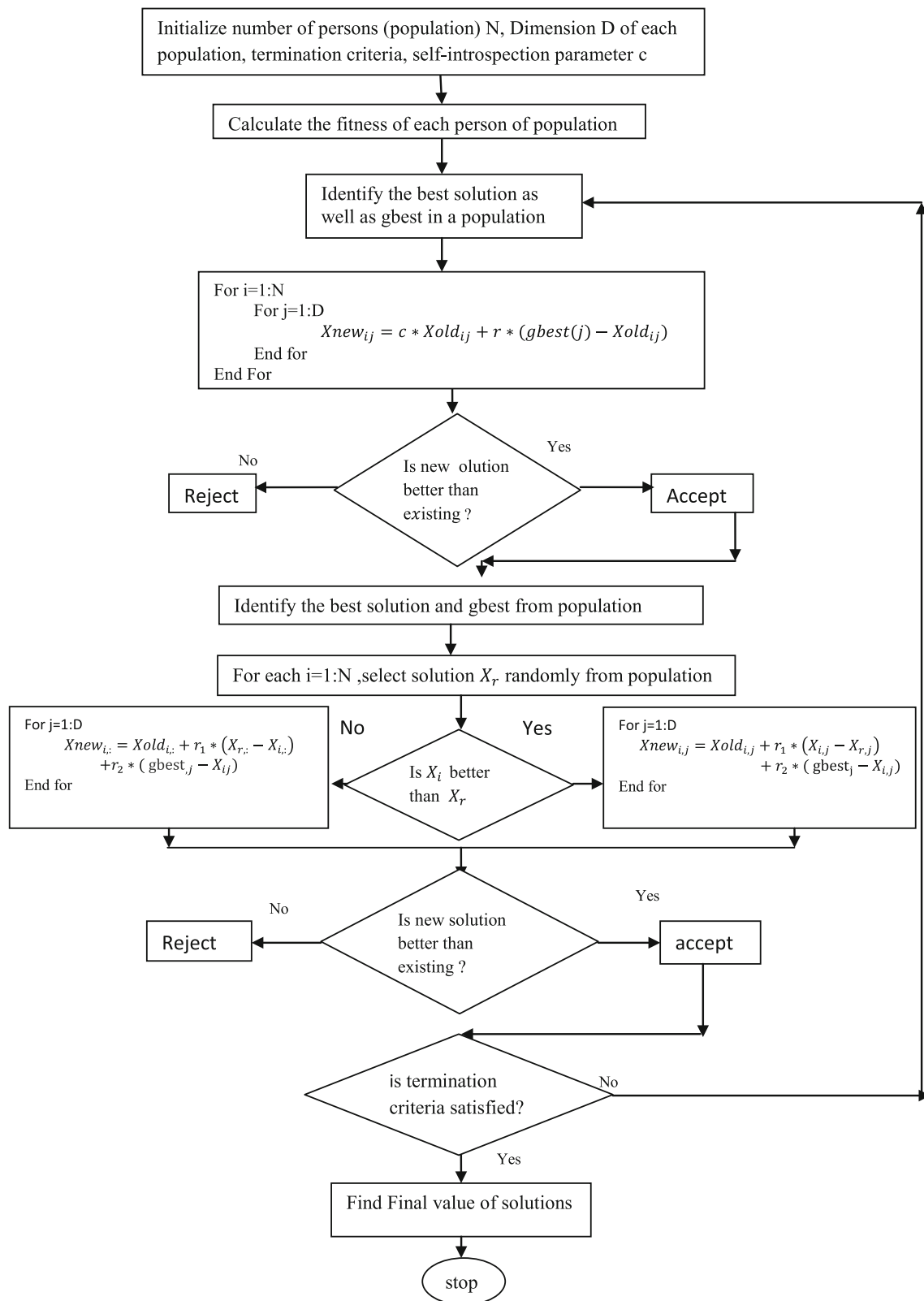
Step 1: *Enumeration of the problem and Initialization of parameters*

Initialize the population size ( $N$ ), number of generations ( $g$ ), number of design variables ( $D$ ), and limits of design variables ( $U_L, L_L$ ). Define the optimization problem as: Minimize  $f(X)$ . Subject to  $= (x_1, x_2, x_3, \dots, x_D)$ , so that  $X_j = (x_{j1}, x_{j2}, x_{j3}, \dots, x_{jD})$ , where  $f(X)$  is the objective function, and  $X$  is a vector for design variables such that  $L_{L,i} \leq x_i \leq U_{L,i}$ .

Step 2: *Initialize the population*

A random population is generated based on the features (number of parameters) and the size of population chosen by user. For SGO, the population size indicates the number of persons and the features indicate the number of traits of a person. This population is articulated as:

$$\text{Population} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,D} \\ \vdots & & & \ddots & \vdots \\ x_{N,1} & x_{N,2} & x_{N,3} & \cdots & x_{N,D} \end{bmatrix}$$



**Fig. 1** Flow Charts of SGO

Calculate the fitness of the population  $f(X)$ .

**Step 3: Improving Phase**

Then, determine  $gbest_g$  using Eq. (1), which is the best solution for that iteration. As in the improving phase, each person gets knowledge from their group's best, i.e.,  $gbest$

For  $i = 1 : N$

For  $j = 1 : D$

$$X_{newij} = c * X_{oldij} + r * (gbest(j) - X_{oldij})$$

End for

End for

The value of  $c$  is self-introspection factor. The value of  $c$  can be empirically chosen for a given problem. We have set it to 0.2 in this work after thorough study of our investigated problems and  $r$  is a random number,  $r \sim U(0, 1)$ .

Accept  $X_{new}$  if it gives better function value.

**Step 4: Acquiring phase**

As explained above, in the acquiring phase, a person of social group interacts with the best person, i.e.,  $gbest$  of the group and also interacts randomly with other persons of the group for acquiring knowledge. The mathematical expression is defined in “Acquiring phase”.

**Step 5: Termination criterion**

Stop the simulation if the maximum generation number is achieved; otherwise, repeat from Steps 3–4.

## Experimental results

In this paper, the performance of SGO is compared with many classical population-based optimization techniques as well as their advanced variants using some basic benchmark functions and 25 test functions proposed in the CEC2005 special session on real-parameter optimization. A description of some basic benchmark functions is given in Appendix, and others are referred from their respective papers, and a detailed description of these 25 CEC2005 test functions can be found in [21]. In “Experiment 1:SGO vs. GA, PSO, DE, ABC, and TLBO” to “Experiment 8. SGO vs FIPS-PSO, CPSO-H, DMSPSO-LS, CLPSO, APSO, SSG-PSO, SSG-PSO-DFP, SSG-PSO-BFGS, SSG-PSO-NM, SSG-PSO-PS”, we have described the experimentation on basic benchmark functions; in “Experiment 9: SGO vs. jDE, SaDE, EPSDE, CoDE, MPEDE, CLPSO, CMA-ES, GL-25 and TLBO”, CEC2005 test functions are experimented; and experiments on composite test functions are discussed and experimented in “Experiment 10: SGO vs. PSO, CPSO, CLPSO, CMA-ES, G3-PCX, DE, and TLBO using Composite functions”. To

have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level was conducted on the experimental results, and the last three rows of each respective table summarize the experimental results.

For comparing the speed of the algorithms, the first thing we require is a fair time measurement. The number of iterations or generations cannot be accepted as a time measure, since the algorithms perform different amount of works in their inner loops, and they have different population sizes. Hence, we choose the number of *fitness function evaluations (FEs)* as a measure of computation time instead of generations or iterations. Since the algorithms are stochastic in nature, the results of two successive runs usually do not match. Hence, we have taken different independent runs (with different seeds of the random number generator) of each algorithm.

Finally, we would like to point out that all the experiment codes are implemented in MATLAB 7. The experiments are conducted on a Pentium 4, 1 GB memory desktop in Windows XP 2002 environment.

### Experiment 1:SGO vs. GA, PSO, DE, ABC, and TLBO

In this section, for fair comparison of the performances of algorithms, the results are directly gained from [22] for GA, PSO, DE and ABC algorithms. However, the simulations have been carried out by us for TLBO and our proposed SGO algorithm. The common parameter such as population size is set to 20 for both TLBO and SGO. The maximum number for function evaluation is set to 2000 for TLBO and 1000 for SGO. The other specific parameters of algorithms are given below:

**TLBO settings** For TLBO, there is no such constant to set.

**SGO settings** For SGO, there is only one constant self-introspection factor for optimum self-effort  $c$ . The value of  $c$  is empirically set to 0.2 for better results.

The 25 benchmark functions which are considered for simulations include many different kinds of problems such as unimodal, multimodal, regular, irregular, separable, non-separable and multidimensional. All problems are divided into four categories such as US, MS, UN, MN, and the range, formulation, characteristics and the dimensions of these problems are described in Appendix.

Each of the experiments for TLBO and SGO is repeated 30 times (we have taken the same number of experimentations which have been done in [22] to make the comparison fair) with different random seeds, and the best mean values produced by the algorithms have been recorded. Comparison criteria are the mean solution and the standard solution for different independent runs. The mean solution describes the average ability of the algorithm to find the global solution, and the standard deviation describes the variation in solution



from the mean solution. To make the comparison clear, the values below  $10^{-12}$  are assumed to be 0. Also, to have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table 1 summarize the results. The results for GA, PSO, DE and ABC are taken from the paper [22]. The Table 1 presents the comparison of fitness values of GA, PSO, DE and ABC gained from [22] and for TLBO and SGO computed by us. In Table 1, "NA" stands for experiment is not conducted for that particular function. The best optimal values are shown in bold face.

From Table 1, it is clear that SGO provides better optimal results in as many as 12 functions compared to GA, 5 functions compared to PSO, 4 functions compared to DE, 3 functions compared to ABC and 2 functions compared to TLBO. It can be arrived at a conclusion here that SGO is very competitive compared to ABC and especially with TLBO. However, we have noted that SGO is faster when compared to other algorithms. It takes 1/500th function evaluations compared to GA, PSO, DE and ABC for all functions except for Rosenbrock wherein it takes 1/50th of total function evaluations. And it is computing almost in half of the total function evaluations compared to TLBO except for Rosenbrock for which SGO takes 1/5th number of function evaluation as against TLBO. From the above findings, we may arrive at a conclusion that our proposed algorithm not only performs better compared to many state-of-the-art algorithms like GA, PSO, DE but also very competitive with ABC and TLBO in providing optimal solutions. Importantly, SGO takes less computation efforts compared to all other algorithms investigated in this section.

### Experiment 2: SGO vs. HS, IBA, ABC, and TLBO

In this experiment, five different benchmark problems from Karaboga and Akay [23] are considered, and comparison is carried out between SGO, the harmony search algorithm (HS), improved bee algorithm (IBA), artificial bee colony optimization (ABC) and teaching–learning-based optimization (TLBO) [24]. To compare the results, the mean solution and the standard solution for different independent runs are taken. In our simulation, we run TLBO for maximum 2000 FEs with 10 as the population size for all functions except Rosenbrock, whereas HS, IBA and ABC run for 50,000 FEs with 50 as population size. For Rosenbrock function, TLBO takes 50,000 FEs with population size of 50. For our optimization algorithm, i.e., SGO, maximum FEs are set to 1000 with 10 as the population size for all functions except Rosenbrock for which it is set to 50,000 FEs with population size of 50. The results are gathered for different independent runs in each case, and the mean and standard deviation are calculated for the results obtained in different runs. Description of the functions is given in Appendix.

In this simulation, different dimensions ( $D$ ) of the benchmark functions are chosen for study. Values starting from as small as 5 to 1000 are taken. The results for dimensions 5, 10, 30, 50, 100 are directly lifted from [24] for all investigated algorithms and put in Table 2. For SGO algorithm, we have computed results for all dimensions and generated results for two large-scale dimensions, such as 500 and 1000, to investigate the performance of SGO for large-dimension problems. The maximum number of FEs for SGO is set 1/50th of maximum FEs taken for HS, IBA and ABC for all functions except Rosenbrock. And, for Rosenbrock, it is 1/5th of HS, IBC and ABC. However, it is exactly half that of TLBO in all functions [24, 25]. It may be emphasized here that the reduced value of maximum number of FEs for SGO is deliberately chosen to investigate the effectiveness and efficiency over other algorithms. Table 2 shows the results for this experiment. In Table, "NA" stands for experiment is not conducted for that particular function. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. It can be seen from Table 2 that SGO has outperformed than all the algorithms for all the functions in almost all dimensions. This experiment shows that SGO is effective in finding the optimum solution with increase in dimensions. However, the performance of other algorithms in higher dimensions has not been ascertained in this work. Our preliminary literature study reveals that they do not perform well in higher dimensions.

### Experiment 3: SGO vs OEA, HPSO-TVAC, CLPSO, APSO, OLPSO-L and OLPSO-G

In this section, comparisons of SGO versus OEA, HPSO-TVAC (Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients) [26], APSO (adaptive particle swarm optimization) [27], CLPSO (comprehensive learning particle swarm optimization) [28], OLPSO (orthogonal learning particle swarm optimization)-L [29] and OLPSO-G [29] on nine benchmarks listed in Appendix are carried out. OEA uses  $3.0 \times 10^5$  FEs, HPSO-TVAC, CLPSO, APSO, OLPSO-L and OLPSO-G use  $2.0 \times 10^5$  FEs, whereas SGO runs for  $3 \times 10^3$  FEs for sphere, schwefel 1.2, schwefel 2.22 function,  $1.0 \times 10^2$  FEs for step,  $4.0 \times 10^2$  FEs for rastrigin, noncontinuous rastrigin and griwank,  $1.0 \times 10^3$  FEs for Ackley and quartic function. The results of OEA, HPSO-TVAC, CLPSO and APSO are gained from [28] and [27] directly, and for OLPSO-L and OLPSO-G, results are gained from [29] directly and put in Table 3. In Table, "NA" stands for experiment is not conducted for that particular function. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum

**Table 1** Performance comparisons of results of 30 runs obtained by GA, PSO, DE, ABC, TLBO, and SGO algorithms in terms of mean and Std

Name of the functions	GA	PSO	DE	ABC	TLBO	SGO
Step						
Mean	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Sphere						
Mean	1.11e+03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(7.41e+01)–	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Sum Squares						
Mean	1.48e+02	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(1.24e+01)–	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Quartic						
Mean	1.81e–01	1.16e–03	1.36e–03	3.00e–02	1.20e–03	3.77e–04
Std	(2.71e–02)–	(2.76e–04)–	(4.17e–04)–	(4.87e–03)–	(3.1134e–04)–	1.46e–04
Beale						
Mean	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Easom						
Mean	–1	–1	–1	–1	–1	–1
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Matyas						
Mean	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	7(0.00e+00)≈	0.00e+00
Zakharov						
Mean	1.36e–02	0.00e+00	0.00e+00	2.48e–04	0.00e+00	0.00e+00
Std	(4.53e–03)–	(0.00e+00)≈	(0.00e+00)≈	(1.83e–04)–	(0.00e+00)≈	0.00e+00
Powell						
Mean	9.70e+00	1.10e–04	2.17e–07	3.13e–03	0.00e+00	0.00e+00
Std	(1.55e+00)–	(1.60e–04)–	(1.36e–07)–	(5.03e–04)–	(0.00e+00)≈	0.00e+00
Schwefel 1.2						
Mean	7.40e+03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(1.14e+03)–	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Schwefel 2.21						
Mean	NA	NA	NA	NA	0.00e+00	0.00e+00
Std					(0.00e+00)≈	0.00e+00
Schwefel 2.22						
Mean	1.1.0e+01	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(1.39e+00)–	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Bohachevsky1						
Mean	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Bohachevsky2						
Mean	6.83e–02	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(7.82e–02)–	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Bohachevsky3						
Mean	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
Booth						
Mean	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00

**Table 1** continued

Name of the functions	GA	PSO	DE	ABC	TLBO	SGO
<b>Rastrigin</b>						
Mean	5.29e+01	4.39e+01	1.17e+01	0.00e+00	0.00e+00	0.00e+00
Std	(4.56e+00)–	(1.17e+01)–	(2.54e+00)–	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
<b>Noncontinuous rastrigin</b>						
Mean	NA	NA	NA	NA	0.00e+00	0.00e+00
Std					(0.00e+00)≈	0.00e+00
<b>Six Hump Camel back</b>						
Mean	–1.0316	–1.0316	–1.0316	–1.0316	–1.0316	–1.0316
Std	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
<b>Griewank</b>						
Mean	1.06e+01	1.74e–02	1.48e–03	0.00e+00	0.00e+00	0.00e+00
Std	(1.16e+00)–	(2.08e–02)–	(2.96e–03)–	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
<b>Ackley</b>						
Mean	1.47e+01	1.65e–01	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Std	(1.78e–01)–	(4.94e–01)–	(0.00e+00)≈	(0.00e+00)≈	(0.00e+00)≈	0.00e+00
<b>Multimod</b>						
Mean	NA	NA	NA	NA	0.00e+00	0.00e+00
Std					(0.00e+00)≈	0.00e+00
<b>Weierstrass</b>						
Mean	NA	NA	NA	NA	0.00e+00	0.00e+00
Std					(0.00e+00)≈	0.00e+00
<b>Elliptic</b>						
Mean	NA	NA	NA	NA	0.00e+00	0.00e+00
Std					(0.00e+00)≈	0.00e+00
<b>Rosenbrocks</b>						
Mean	1.96e+05	1.51e+01	1.82e+01	8.88e–02	2.71e+01	2.70e+01
Std	(3.85e+04)–	(2.411e+01)+	(5.03e+00)+	(7.74e–02)+	(1.14e+00)–	1.76e–01
–	12	05	04	03	02	
+	00	01	01	01	00	
≈	08	14	15	16	23	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of GA, PSO, DE, ABC and TLBO. “–”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. According to Wilcoxon's rank-sum test, SGO performs superior than OEA in seven test functions and comparable to one test function out of eight test functions, improved than HPSO-TVAC in eight test functions and equivalent to one test function out of nine test functions. It is also found to be better than APSO in eight test functions and equivalent to one test function out of nine test functions. In our work, we observe that compared to OLPSO-L, our proposed SGO is better in two OLPSO-G in four test functions and equivalent to one test function out of five test functions. Hence, it can be claimed that even though the maximum number of fitness evaluations for SGO is less than the other

algorithms, still SGO is either better than or equivalent to other algorithms for each benchmark function according to the Wilcoxon's rank-sum test.

#### Experiment 4: SGO vs JADE, jDE, SaDE, CoDE and EPSDE

The experiments in this section constitute the comparison of the SGO algorithm versus SaDE [30], jDE [31], JADE [32], CoDE [33] and EPSDE [34] on nine benchmark functions which are listed in Appendix. The results of JADE, jDE and SaDE are gained from [32] directly and put in Table 4. For CoDE and EPSDE, we have generated results using codes given in website Q. Zhang's homepage: <http://dces.essex.ac>.



**Table 2** Performance comparisons of results of 30 runs obtained by HS, IBA, ABC, TLBO and SGO algorithms in terms of mean and Std

Name of function	$D$	HS Mean	IBA Mean	ABC Mean	TLBO Mean	SGO Mean
Sphere	5	3.20e−10	3.91e−17	4.30e−17	5.03e−33	1.58e−67
		(2.89e−10)−	(1.24e−17)−	(1.07e−17)−	(1.27e−32)−	1.93e−67
	10	6.45e−08	4.95e−17	7.36e−17	2.26e−29	2.01e−66
		(3.07e−08)−	(2.30e−17)−	(4.43e−17)−	(3.61e−29)−	7.99e−67
	30	7.21e+00	2.92e−16	4.69e−16	6.90e−26	9.19e−66
		(3.62e+00)−	(6.77e−17)−	(1.07e−16)−	(3.18e−25)−	9.21e−67
	50	5.46e+02	5.39e−16	1.19e−15	8.71e−26	1.66e−65
		(1.78e+03)−	(1.07e−16)−	(4.68e−16)−	(1.86e−25)−	1.80e−66
	100	1.90e+04−	1.45e−15	1.99e−06	9.42e−26	3.65e−65
		(1.78e+03)−	(1.63e−16)−	(2.26e−06)−	(3.70e−25)−	1.58e−66
Rosenbrock	500	NA	NA	NA	NA	1.96e−64
						9.88e−66
	1000	NA	NA	NA	NA	4.01e−64
						1.69e−65
	5	5.94e+00	4.55e−01	2.33e−01	1.80e−01	4.20e−06
		(6.71e+00)−	(1.54e+00)−	(2.24e−01)−	(8.04e−02)−	2.30e−06
	10	6.52e+00	1.10e+01	4.62e−01	5.58e+00	4.15e−02
		(8.16e+00)−	(2.55e+01)−	(5.44e−01)−	(6.18e−01)−	1.82e−01
	30	3.82e+02	7.57e+01	9.98e−01	2.71e+01	2.30e+01
		(5.29e+02)−	(1.16e+02)−	(1.52e+00)+	(1.14e+00)−	6.76e−01
Ackley	50	2.47e+04	6.30e+02	4.33e+00	4.78e+01	4.42e+01
		(1.02e+04)−	(1.20e+03)−	(5.48e+00)+	(1.01e+00)−	5.01e−01
	100	1.45e+07	6.42e+02	1.12e+02	9.81e+01	9.50e+01
		(2.16e+06)−	(8.20e+02)−	(6.92e+1)−	(3.61e−01)−	6.01e−01
	500	NA	NA	NA	NA	4.92e+02
						6.67e−01
	1000	NA	NA	NA	NA	9.89+02
						2.12e−01
	5	2.68e−05	6.35e−10	9.64e−17	0.00e+00	−8.88e−16
		(1.24e−05)−	(9.77e−11)−	(5.24e−17)+	(0.00e+00)+	0.00e+00
Ackley	10	2.76e−04	6.71e−02	3.51e−16	0.00e+00	−8.88e−16
		(7.58e−05)−	(3.61e−01)−	(6.13e−17)≈	(0.00e+00)+	0.00e+00
	30	9.43e−01	1.75e+00	3.86e−15	7.11e−16	0.00e+00
		(5.63e−01)−	(9.32e−01)−	(3.16e−15)−	(1.82e−15)≈	0.00e+00
	50	5.28e+00	8.43e+00	4.38e−08	1.24e−15	−8.88e−16
		(4.03e−01)−	(7.70e+00)−	(4.65e−08)−	(1.95e−15)−	0.00e+00
	100	1.32e+01	1.89e+01	1.32e−02	2.13e−15	−8.88e−16
		(4.90e−01)−	(8.50e−01)−	(1.30e−02)−	(1.19e−15)−	0.00e+00
	500	NA	NA	NA	NA	−8.88e−16
						0.00e+00
Ackley	1000	NA	NA	NA	NA	−8.88e−16
						0.00e+00

**Table 2** continued

Name of function	<i>D</i>	HS Mean	IBA Mean	ABC Mean	TLBO Mean	SGO Mean
Griwank	5	2.60e−02	3.14e+00	4.04e−17	0.00e+00	0.00e+00
		(1.38e−02)−	(1.41e+00)−	(1.12e−17)−	(0.00e+00)≈	0.00e+00
	10	1.02e+00	1.04e+00	6.96e−17	0.00e+00	0.00e+00
		(3.02e−02)−	(1.13e+00)−	(4.06e−17)−	(0.00e+00)≈	0.00e+00
	30	1.09e+00	6.68e+00	5.82e−06	0.00e+00	0.00e+00
		(3.92e−02)−	(6.43e+00)−	(3.13e−05)−	(0.00e+00)≈	0.00e+00
	50	5.81e+00	1.34e+02	5.72e−01	0.00e+00	0.00e+00
		(9.16e−01)−	(2.41e+01)−	(9.22e−01)−	(0.00e+00)≈	0.00e+00
	100	1.78e+02	7.93e+02	1.31e+01	0.00e+00	0.00e+00
		(1.98e+01)−	(7.96e+01)−	(6.30e+00)−	(0.00e+00)≈	0.00e+00
	500	NA	NA	NA	NA	0.00e+00
	1000	NA	NA	NA	NA	0.00e+00
Rastrigin	5	6.07e−08	4.58e+00	4.34e−17	0.00e+00	0.00e+00
		(5.52e−08)−	(2.31e+00)−	(1.10e−17)−	(0.00e+00)≈	0.00e+00
	10	1.05e−05	2.20e+01	5.77e−17	0.00e+00	0.00e+00
		(5.23e−06)−	(7.46e+00)−	(2.98e−17)−	(0.00e+00)≈	0.00e+00
	30	7.40e−01	1.28e+02	4.80e−05	0.00e+00	0.00e+00
		(7.00e−01)−	(2.49e+01)−	(2.43e−04)−	(0.00e+00)≈	0.00e+00
	50	3.76e+01	2.72e+02	4.72e−01	0.00e+00	0.00e+00
		(4.87e+00)−	(3.27e+01)−	(4.92e−01)−	(0.00e+00)≈	0.00e+00
	100	3.15e+02	6.49e+02	1.46e+01	0.00e+00	0.00e+00
		(2.33e+01)−	(4.52e+01)−	(4.18e+00)−	(0.00e+00)≈	0.00e+00
	500	NA	NA	NA	NA	0.00e+00
	1000	NA	NA	NA	NA	0.00e+00
−		25	25	21	12	0.00e+00
+		00	00	3	02	
≈		00	00	01	11	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of HS, IBA, ABC, and TLBO. "−", "+", and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

*D* dimension

[uk/staff/qzhang/](http://uk/staff/qzhang/). For CoDE, EPSDE and SGO, we have considered population size as 20. The maximum number of fitness evaluations for each function is different, and FEs are noted in bracket of each cell in Table 4. Fitness values are shown in Table 4 in means and standard deviations. In Table, "NA" stands for experiment is not conducted for that particular function. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. According to Wilcoxon's rank-sum test, it can be noted that the performance of SGO is always better than all other algorithms except EPSDE in reporting

the optimal value, where SGO performs better than EPSDE in five test functions and equivalent to three test functions out of eight test functions. So, it is interesting to note that even though the maximum number of fitness evaluations for SGO is less than the other algorithms, still SGO is better than or equivalent with all variants of DE algorithm in this experiment according to Wilcoxon's rank-sum test.

### Experiment 5: SGO vs. CABC, GABC, RABC and IABC

In this section, we compare SGO with CABC [35], GABC [36], RABC [37] and IABC [38] on eight benchmark func-

**Table 3** Performance comparisons of SGO, OEA, HPSO-TVAC, CLPSO, APSO, OLPSO-L and OLPSO-G

Function	OEA	HPSO-TVAC	CLPSO	APSO	OLPSO-L	OLPSO-G	SGO
Sphere							
Mean	2.48e−30	3.38e−41	1.89e−19	1.45e−150	1.11e−38	4.12e−54	1.46e−205
Std	(1.13e−29)−	(8.50e−41)−	(1.49e−19)	(5.73e−150)−	(1.28e−38)−	(6.34e−54)−	0
Schwefel 2.22							
Mean	2.07e−13	6.9e−23	1.01e−13	5.15e−84	7.67e−22	9.85e−30	1.85e−103
Std	(2.44e−12)−	(6.89e−23)−	(6.54e−14)−	(1.44e−83)−	(5.63e−22)−	(1.01e−29)−	2.03e−104
Schwefel 1.2							
Mean	1.88e−09	2.89e−07	3.97e+02	1.0e−10	NA	NA	4.31e−201
Std	(3.726e−9)−	(2.97e−07)−	(1.42e+02)−	(2.13e−10)−			0
Step							
Mean	0	0	0	0	NA	NA	0
Std	(0)≈	(0)≈	(0)≈	(0)≈			0
Rastrigin							
Mean	5.43e−17	2.39 e+00−	2.57e−11	5.8e−15	0	1.07e+00	0
Std	(1.68e−16)−	(3.71e+00)−	(6.64e−11)−	(1.01e−14)−	(0)≈	(9.90e−01)−	0
Noncontinuous rastrigin							
Mean	NA	1.83e+00	1.67e−01	4.14e−16	NA	NA	0
Std		(2.65e+00)−	(3.79e−01)−	(1.45e−15)−			0
Ackley							
Mean	5.35e−14	2.06e−10	2.01e−12	1.11e−14	4.14e−15	7.98e−15	−8.88e−16
Std	(2.94e−13)−	(9.45e−10)−	(9.22e−13)−	(3.55e−15)−	(0)≈	(2.03e−15)≈	1.01e−31
Griewank							
Mean	1.32e−02	1.07e−02	6.45e−13	1.67e−02	0	4.83e−03	0
Std	(1.56e−02)−	(1.14e−02)−	(2.07e−12)−	(2.41e−02)−	(0)≈	(8.63e−03)−	0
Quartic							
Mean	3.29e−03	5.54e−02	3.92e−03	4.66e−03	NA	NA	5.37e−04
Std	(1.09e−03)−	(2.08e−02)−	(1.14e−03)−	(1.70e−03)−			3.91e−05
−	7	8	8	8	2	4	
+	00	00	00	00	00	00	
≈	1	1	1	1	3	1	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of OEA, HPSO-TVAC, CLPSO, APSO, OLPSO-L, OLPSO-G. "−", "+", and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

tions. The parameters of the algorithms are identical to [36]. The maximum number of fitness evaluations for each function is different, and FEs are noted in bracket of each cell in Table 5. The results have been summarized in Table 5. The fitness value in terms of mean and standard deviation is reported. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. It can be observed from Table 5 that SGO performs better in comparison to all algorithms. So, we can say that even though the maximum number of fitness evaluations for SGO is less than the other algorithms, still SGO is better than all variants of ABC algorithm in this experiment.

## Experiment 6: SGO vs. TLBO

In this experiment, we compare only TLBO and SGO algorithms. As TLBO is relatively new compared to other algorithms investigated in our work, we have devoted a special section to compare our approach with TLBO. In this experiment, our main objective is to see how SGO performs against TLBO in terms of optimal solution and computational costs. The common parameter such as population size is taken as 20 and maximum number fitness function evaluation is taken as 1,000 for both TLBO and SGO.

We used 25 benchmark problems to test the performance of the TLBO and our proposed SGO algorithms. The ini-

**Table 4** Performance comparisons of SGO, JADE, jDE, SaDE, CoDE and EPSDE

Function	SaDE (FEs)	jDE (FEs)	JADE (FEs)	CoDE (FEs)	EPSDE (FEs)	SGO (FEs)
Sphere						
Mean	4.5e–20 ( $1.5 \times 10^5$ )	2.5e–28 ( $1.5 \times 10^5$ )	1.8e–60 ( $1.5 \times 10^5$ )	1.12e–31 ( $1.5 \times 10^5$ )	1.53e–85 ( $1.5 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(1.90e–14)–	(3.5e–28)–	(8.4e–60)–	(3.45e–31)–	(9.01e–86)–	0
Schwefel 2.22						
Mean	1.9e–14 ( $2.0 \times 10^5$ )	1.5e–23 ( $2.0 \times 10^5$ )	1.8e–25 ( $2.0 \times 10^5$ )	1.22e–23 ( $2.0 \times 10^5$ )	3.18e–54 ( $2.0 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(1.10e–14)–	(1.0e–23)–	(8.8e–25)–	(3.90e–23)–	(3.11e–54)–	0
Schwefel 1.2						
Mean	9.0e–37 ( $5.0 \times 10^5$ )	5.2e–14 ( $5.0 \times 10^5$ )	5.7e–61 ( $5.0 \times 10^5$ )	7.86e–31 ( $5.0 \times 10^5$ )	4.81e–76 ( $5.0 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(5.40e–36)–	(1.1e–13)–	(2.7e–60)–	(1.86e–32)–	(1.90e–76)–	0
Step						
Mean	9.3e+02 ( $1.0 \times 10^4$ )	1.0e+03 ( $1.0 \times 10^4$ )	2.9e+00 ( $1.0 \times 10^4$ )	3.00e+00 ( $1.0 \times 10^4$ )	0 ( $1.0 \times 10^4$ )	0 ( $1.0 \times 10^2$ )
Std	(1.8e+02)–	(2.2e+02)–	(1.2e+00)–	(1.90e+00)–	(0)≈	0
Rastrigin						
Mean	1.2e–03 ( $1.0 \times 10^5$ )	1.5e–04 ( $1.0 \times 10^5$ )	1.0e–04 ( $1.0 \times 10^5$ )	1.21e–01 ( $1.0 \times 10^5$ )	0 ( $1.0 \times 10^5$ )	0 ( $4.0 \times 10^2$ )
Std	(6.5e–04)–	(2.0e–04)–	(6.0e–05)–	(3.89e–02)–	(0)≈	0
Schwefel 2.21						
Mean	7.4e–11 ( $5.0 \times 10^5$ )	1.4e–15 ( $5.0 \times 10^5$ )	8.2e–24 ( $5.0 \times 10^5$ )	2.44e–27 ( $5.0 \times 10^5$ )	1.94e–2 ( $5.0 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(1.82e–10)–	(1.0e–15)–	(4.0e–23)–	(1.89e–27)–	(8.90e–4)–	0
Ackley						
Mean	2.7e–03 ( $5.0 \times 10^4$ )	3.5e–04 ( $5.0 \times 10^4$ )	8.2e–10 ( $5.0 \times 10^4$ )	1.18e–04 ( $5.0 \times 10^4$ )	5.36e–13 ( $5.0 \times 10^4$ )	4.19e–14 ( $1.0 \times 10^3$ )
Std	(5.1e–04)–	(1.0e–04)–	(6.9e–10)–	(4.90e–04)–	(4.77e–14)–	2.85e–14
Griewank						
Mean	7.8e–04 ( $5.0 \times 10^4$ )	1.9e–05 ( $5.0 \times 10^4$ )	9.9e–08 ( $5.0 \times 10^4$ )	1.74e–07 ( $5.0 \times 10^4$ )	0 ( $5.0 \times 10^4$ )	0 ( $4.0 \times 10^2$ )
Std	(1.2e–03)–	(5.8e–05)–	(6.0e–07)–	(2.33e–07)–	(0)≈	0
Quartic						
Mean	4.8e–03 ( $3.0 \times 10^5$ )	3.3e–03 ( $3.0 \times 10^5$ )	6.4e–04 ( $3.0 \times 10^5$ )	NA	NA	5.37e–04 ( $1.0 \times 10^3$ )
Std	(1.2e–03)–	(8.5e–04)–	(2.5e–04)–			3.91e–05
–	9	9	9	8	5	
+	00	00	00	00	00	
≈	00	00	00	00	3	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of JADE, jDE, SaDE, CoDE and EPSDE. “–”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

**Table 5** Performance comparisons of SGO, CABC, GABC, RABC and IABC

Function	CABC	GABC	RABC	IABC	SGO
Sphere					
Mean	2.3e−40 ( $1.5 \times 10^5$ )	3.6e−63 ( $1.5 \times 10^5$ )	9.1e−61 ( $1.5 \times 10^5$ )	5.34e−178 ( $1.5 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(1.7e−40)−	(5.7e−63)−	(2.1e−60)−	(0)−	0
Schwefel 2.22					
Mean	3.5e−30 ( $2.0 \times 10^5$ )	4.8e−45 ( $2.0 \times 10^5$ )	3.2e−74 ( $2.0 \times 10^5$ )	8.82e−127 ( $2.0 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(4.8e−30)−	(1.4e−45)−	(2.0e−73)−	(3.49e−126)−	0
Schwefel 1.2					
Mean	8.4e+02 ( $5.0 \times 10^5$ )	4.3e+02 ( $5.0 \times 10^5$ )	2.9e−24 ( $5.0 \times 10^5$ )	1.78e−65 ( $5.0 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(9.1e+02)−	(8.0e+02)−	(1.5e−23)−	(2.21e−65)−	0
Step					
Mean	0 ( $1.0 \times 10^4$ )	0 ( $1.0 \times 10^4$ )	0 ( $1.0 \times 10^4$ )	0 ( $1.0 \times 10^4$ )	0 ( $1.0 \times 10^2$ )
Std	0≈	0≈	0≈	0≈	0
Rastrigin					
Mean	1.3e−00 ( $5.0 \times 10^4$ )	1.5e−10 ( $5.0 \times 10^4$ )	2.3e−02 ( $5.0 \times 10^4$ )	0 ( $5.0 \times 10^4$ )	0 ( $4.0 \times 10^2$ )
Std	(2.7e−00)−	(2.7e−10)−	(5.1e−01)−	0≈	0
Schwefel 2.21					
Mean	6.1e−03 ( $5.0 \times 10^5$ )	3.6e−06 ( $5.0 \times 10^5$ )	2.8e−02 ( $5.0 \times 10^5$ )	4.98e−38 ( $5.0 \times 10^5$ )	0 ( $5.0 \times 10^3$ )
Std	(5.7e−03)−	(7.6e−07)−	(1.7e−02)−	(8.59e−38)−	0
Ackley					
Mean	1.0e−05 ( $5.0 \times 10^4$ )	1.8e−09 ( $5.0 \times 10^4$ )	9.6e−07 ( $5.0 \times 10^4$ )	3.87e−14 ( $5.0 \times 10^4$ )	2.47e−15 ( $1.0 \times 10^3$ )
Std	(2.4e−06)−	(7.7e−10)−	(8.3e−07)−	(8.52e−15)−	1.82e−15
Griewank					
Mean	1.2e−04 ( $5.0 \times 10^4$ )	6.0e−13 ( $5.0 \times 10^4$ )	8.7e−08 ( $5.0 \times 10^4$ )	0 ( $5.0 \times 10^4$ )	0 ( $4.0 \times 10^2$ )
Std	(4.6e−04)−	(7.7e−13)−	(2.1e−08)−	0≈	0
−	07	07	07	05	
+	00	00	00	00	
≈	01	01	01	04	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of CABC, GABC, RABC and IABC. “−”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

tial range, formulation, characteristics and the dimensions of these problems are listed in Appendix. Each simulation runs for 30 times. The simulation is terminated on attaining maximum number of evaluations or obtaining global minimum values with different random seeds. The mean value and standard deviations of fitness value produced by the algorithms have been recorded in Table 6. At the same time, mean value and standard deviations of number of fitness evaluation produced by the algorithms have also been recorded in Table 7. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. It is observed that SGO performs better in 23 test functions and equivalent in 2 test functions.

From Tables 6 and 7, it is clear that except step and six-hump camel-back function, in all cases, SGO has shown better result than TLBO, and the maximum number of fit-

ness evaluations for easom, bohachevsky1, bohachevsky2, bohachevsky3, rastrigin, noncontinuous rastrigin, multimod and for weierstrass function is less than that of TLBO algorithm and all these functions have reached to optimal solution. In both step and six-hump camel-back function cases, both TLBO and SGO algorithms have performed equivalently and given optimal result, however, in both cases, SGO reaches optimum value with lesser number of fitness evaluations than TLBO. So, we can say that SGO is better than TLBO algorithm in all cases in this experiment. The convergence characteristics of both algorithms have been shown in the graphs below (Fig. 2).

#### Experiment 7: SGO vs. SAABC, GABC, IABC, ABC/Best1, GPSO, DBMPSO, TCP SO and VABC

In this section, we compare SGO with both ABC and PSO variants of algorithm such as GABC(Gbest-guided artificial

**Table 6** Performance comparisons of SGO and TLBO

Function no.	TLBO	SGO	TLBO	SGO
Step			Bohachevsky1	
Mean	0	0	Mean	6.97e−06
Std	(0)≈	0	Std	(3.78e−06)−
Sphere			Bohachevsky2	
Mean	3.20e−03	7.82e−031	Mean	5.16e−06
Std	(2.50e−03)−	1.20e−031	Std	(3.44e−06)−
Sum Squares			Bohachevsky3	
Mean	1.20e−03	1.12e−31	Mean	3.13e−06
Std	(1.50e−03)−	1.25e−33	Std	(2.78e−06)−
Quartic			Booth	
Mean	1.46 e−02	2.98e−04	Mean	1.20e−09
Std	(8.90e−03)−	5.67e−05	Std	(1.17e−09)−
Beale			Rastrigin	
Mean	2.87e−08	2.11e−09	Mean	1.86e+02
Std	(1.24e−09)−	3.88e−11	Std	(3.50e+01)−
Easom			Noncontinuous rastrigin	
Mean	−9.94e−01	−1	Mean	1.61e+02
Std	3.40e−03	−0	Std	(2.69e+01)−
Matyas			Six Hump Camel Back	
Mean	8.08e−12	4.91e−41	Mean	−1.0316
Std	(5.22e−12)−	6.83e−42	Std	(6.79e−16)−
Zakharov			Griewank	
Mean	7.29e−05	1.24e−33	Mean	1.14e−02
Std	(4.98e−05)−	1.92e−35	Std	(9.50e−03)−
Powell			Ackley	
Mean	6.41e−04	3.72e−32	Mean	3.14e−02
Std	(3.37e−04)−	4.08e−34	Std	(3.11e−02)−
Schwefel 1.2			Multimod	
Mean	3.62e+02	1.15e−26	Mean	9.34e−49
Std	(1.34e+02)−	2.67e−29	Std	(1.31e−48)−
Schwefel 2.21			Weierstass	
Mean	5.31e−02	2.05e−16	Mean	5.18e−01
Std	(5.60e−03)−	5.90e−19	Std	(1.89e−01)−
Schwefel 2.22			Elliptic	
Mean	2.25e−02	4.56e−16	Mean	6.69e+02
Std	(2.13e−02)−	1.91e−19	Std	(4.56e+02)−
			Rosenbrocks	
			Mean	29.1979
			Std	(1.1425)−
−	11		−	12
+	00		+	00
≈	01		≈	01

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and TLBO. “−”, “+”, and “≈” denote that the performance of the TLBO algorithm is worse than, better than, and similar to that of SGO, respectively

bee colony algorithm) [36], IABC [38], ABC/Best1 [39], SAABC(simulated annealing-based artificial bee colony) [40], VABC(velocity-based artificial bee colony algorithm)

[41], CPSO(Chaotic particle swarm optimization) [42], DBMPSO(particle swarm optimization with double-bottom chaotic maps) [43], TCPSO(two-swarm cooperative particle



**Table 7** Fitness comparisons of SGO and TLBO

Function no.	TLBO	SGO	TLBO	SGO
Step			Bohachevsky1	
Mean	880	160	Mean	1000
Std	80	40	Std	1000
Sphere			Bohachevsky2	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Sum Squares			Bohachevsky3	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Quartic			Booth	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Beale			Rastrigin	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Easom			Noncontinuous rastrigin	
Mean	1000	640	Mean	1000
Std	0	0	Std	0
Matyas			Six Hump Camel Back	
Mean	1000	1000	Mean	680
Std	0	0	Std	80
Zakharov			Griewank	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Powell			Ackley	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Schwefel 1.2			Multimod	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Schwefel 2.21			Weierstass	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
Schwefel 2.22			Elliptic	
Mean	1000	1000	Mean	1000
Std	0	0	Std	0
			Rosenbrocks	
			Mean	1000
			Std	0

swarms optimization) [44] on 23 benchmark functions out of which 15 are multidimensional and 8 are fixed-dimensional benchmark functions. All functions are described in [41]. The maximum number of fitness evaluations is taken as 40,000, and population size is 40, and the parameters of the algorithms are identical to [41]. The results of SAABC, GABC,

IABC, ABC/Best1, VABC, CPSO, DBMPSO and TCP SO are gained from [41] for comparison with SGO. The comparison results are shown in Tables 8, 9, 10 in terms of means and standard deviations (Std) of the solutions in the 30 independent runs. Tables 8, 9 show the results for 60 and 100 dimensions, respectively, on multidimensional functions, and Table 10 reports the results on the fixed-dimensional functions. The best optimal values are shown in bold face.

As seen from the Tables 8, 9 results, SGO found the global optimal values for all the functions except  $F_6$ ,  $F_7$ ,  $F_9$ ,  $F_{12}$ ,  $F_{14}$  and  $F_{15}$  function for both 60 and 100 dimension. On the other hand, for test functions  $F_9$ ,  $F_{12}$ ,  $F_{14}$  and  $F_{15}$ , the objective values obtained by SGO are extremely close to global optima. Again, as seen from Table 10 results, SGO found the global optimal values for the functions  $F_{16}$ ,  $F_{17}$  and  $F_{21}$  of fixed-dimensional functions. On the other hand, for test functions  $F_{19}$  and  $F_{20}$ , the objective values obtained by SGO are extremely close to global optima.

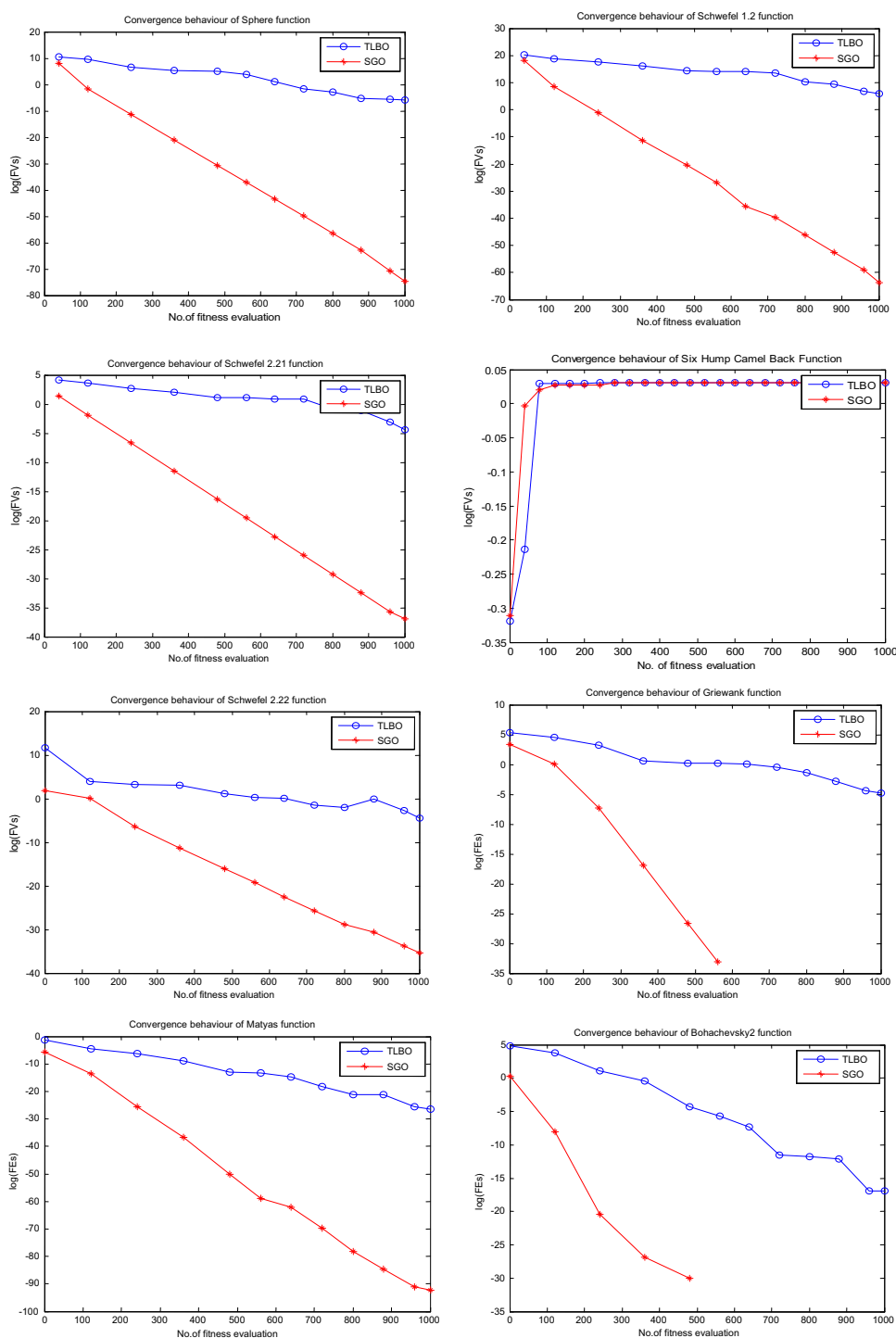
To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. In 60-dimensional case, according to Wilcoxon's rank-sum test, SGO performs superior than SAABC in 14 test functions and comparable to one test function out of a 5 test functions, improved than GABC in all 15 test functions. It is also found to be better than IABC, ABC/Best1, CPSO, DBMPSO and TCP SO in all 15 test functions. In our work, we observe that compared to VABC, our proposed SGO is better in 11 test functions and equivalent to 2 test functions out of 15 test functions.

In 100-dimensional case, according to Wilcoxon's rank-sum test, SGO performs superior than all algorithms except VABC algorithm in all 15 multidimensional test functions. Compared to VABC, our proposed SGO is better in 12 test functions and equivalent to 1 test functions out of 15 test functions.

In fixed-dimensional case, according to Wilcoxon's rank-sum test, SGO performs superior than SAABC in four test functions and equivalent with three test functions out of eight test functions, improved than GABC in five test functions and equivalent with two test function out of eight test functions. It is also found to be better than IABC, ABC/Best1, VABC, CPSO, DBMPSO, TCP SO in seven, five, three, two, three and seven test functions, respectively, out of eight test functions and similarly equivalent with one, two, four, five, four and zero test functions, respectively, out of eight test functions.

So, it is interesting to note that the performance of SGO is better than other algorithms according to Wilcoxon's rank-sum test.

**Fig. 2** Convergence characteristics of SGO Vs TLBO

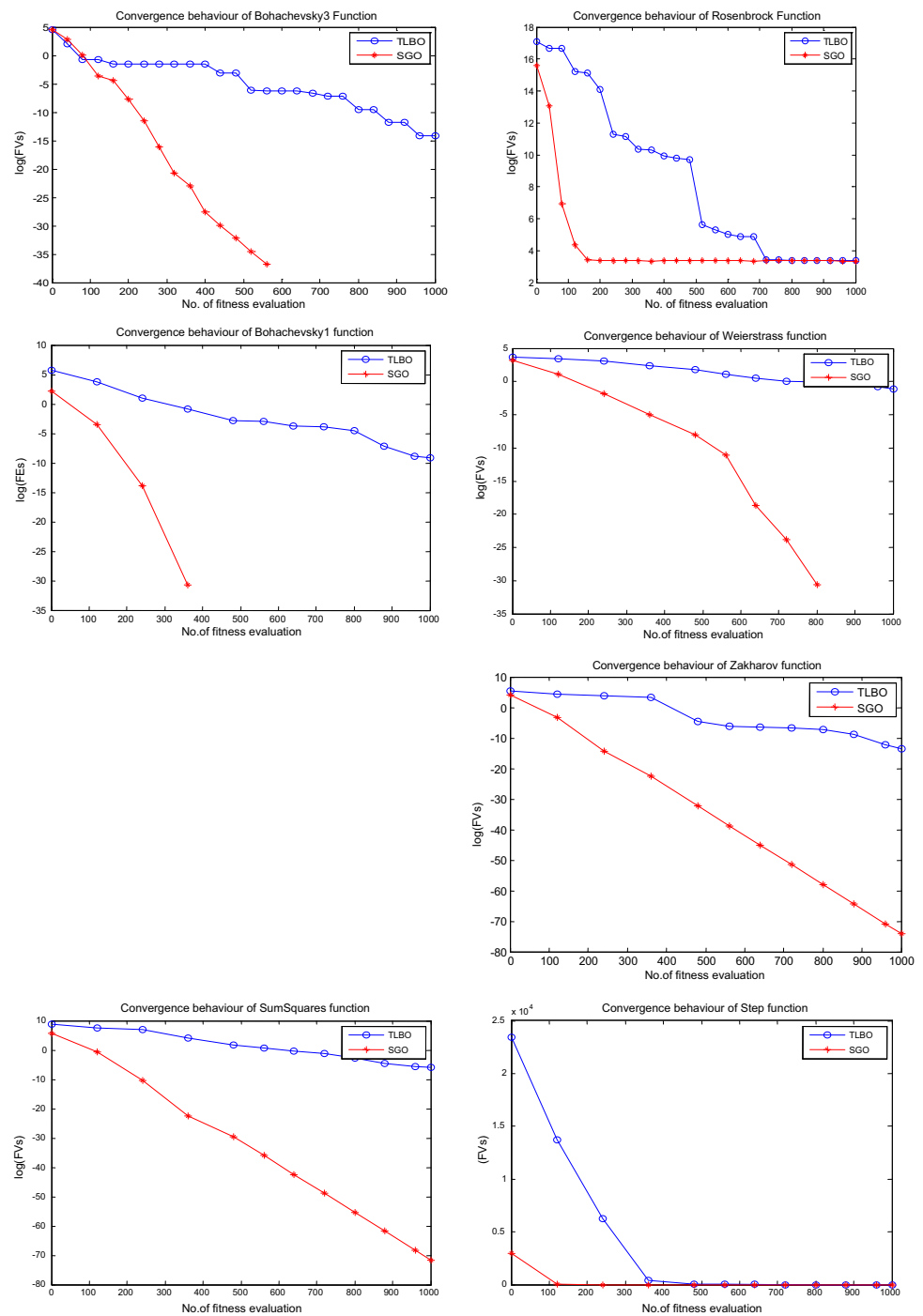


### Experiment 8: SGO vs. GPSO, LPSO, FIPS, SPSSO, CLPSO, OLPSO and SLPSOA

In this section, the comparison of SGO versus GPSO(global PSO) [45], LPSO(local PSO) [46], FIPS(fully informed particle swarm) [47], SPSSO(standard for particle swarm

optimization) [48], CLPSO [28], OLPSO [29] and CLPSOA(scatter learning particle swarm optimization Algorithm) [49], on 14 benchmark functions described in paper [49]. The maximum number of fitness evaluations is taken as 200,000, and population size is 40, and the parameters of the algorithms are identical to [49]. The comparison

Fig. 2 continued

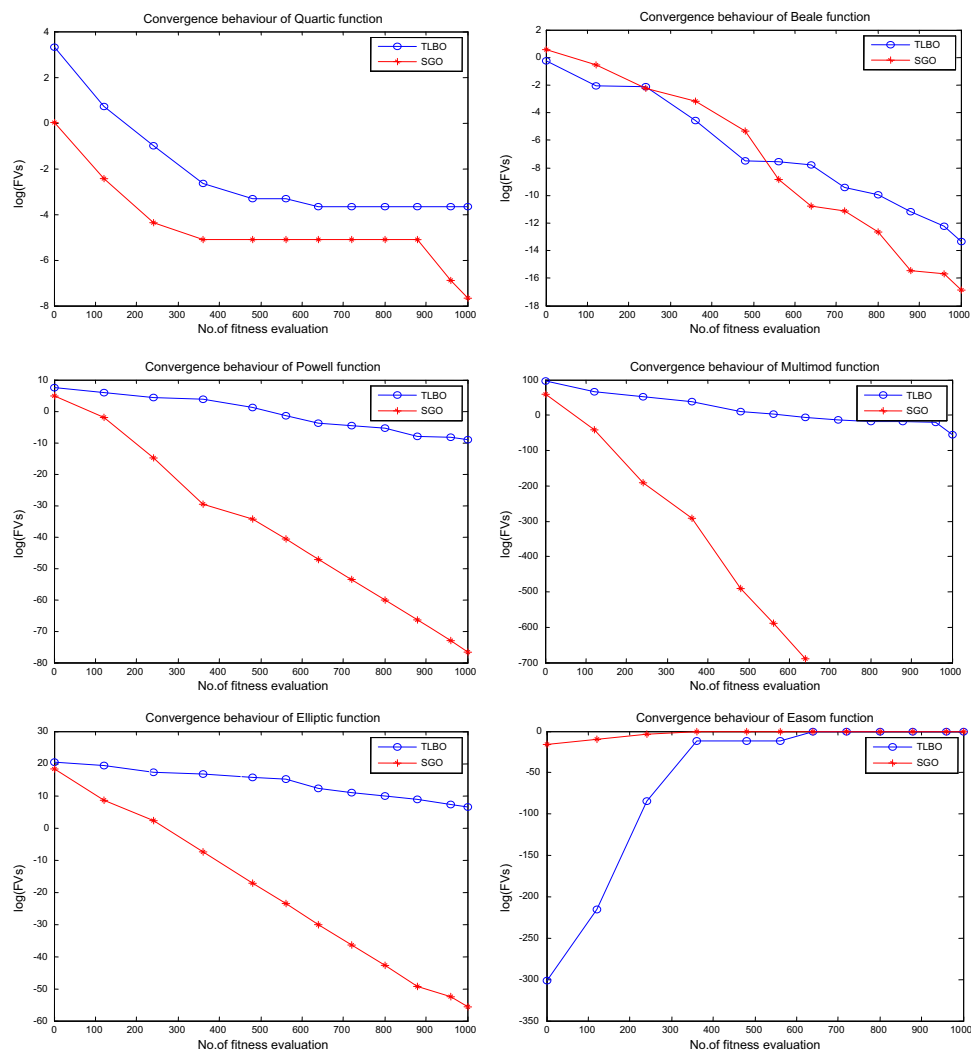


results are shown in Table 11 in terms of means and standard deviations (Std) of the solutions in the 25 independent runs. The results of GPSO, LPSO, FIPS, SPSO, CLPSO, OLPSO and SLPSOA are gained from [49]. As seen from Table 11 results, SGO found the global optimal solution for the functions sphere, schwefel 2.22, rastrigin, griewank, rotated rastrigin and rotated griewank. On the other hand, for the test functions noise, Ackley, generalized penalized, generalized penalised1, and rotated Ackley, the

objective values obtained by SGO are extremely close to global optima. The best optimal values are shown in bold face.

To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. According to Wilcoxon's rank-sum test, SGO performs superior than GPSO, LPSO, FIPS, SPSO, CLPSO, OLPSO and SLPSOA in 13, 14, 13, 12, 12, 6, and 7

Fig. 2 continued



test functions, respectively, out of 14 test functions. The SGO algorithm is equivalent with OLPSO in four test functions and with SLPSOA in two test functions. So, it is interesting to tell according to this experiment that SGO is better than other algorithms according to Wilcoxon's rank-sum test.

#### Experiment 8. SGO vs FIPS-PSO, CPSO-H, DMSPSO-LS, CLPSO, APSO, SSG-PSO, SSG-PSO-DFP, SSG-PSO-BFGS, SSG-PSO-NM, SSG-PSO-PS

To comprehensively compare the performance of SGO with the performance of SSG-PSO (Superior solutions guided PSO with the individual level based mutation operator) and its several variants with different local search techniques, 21 test benchmark functions of different types were used, including unimodal functions, multimodal functions, miscaled functions and rotated functions. The detailed infor-

mation of test functions is displayed in [50]. Here, the popular PSO variants are FIPS-PSO (fully informed PSO), CPSO-H (cooperative based PSO), DMSPSO-LS (dynamic multi-swarm particle swarm optimizer with local search), CLPSO (comprehensive learning PSO), APSO (adaptive particle swarm optimization), and different variants of SSG-PSO with different local search techniques are SSG-PSO-DFP (SSG-PSO with Davidon–Fletcher–Powell method), SSG-PSO-BFGS (SSG-PSO with Broyden–Fletcher–Goldfarb–Shanno method), SSG-PSO-NM (SSG-PSO with Nelder–Mead simplex search), SSG-PSO-PS (SSG-PSO with Pattern Search).

The maximum number of fitness evaluations is taken as 300,000, and population size is 40, and the parameters of the algorithms are identical to [50]. The comparison results are shown in Table 12 in terms of means and standard deviations (Std) of the solutions in the 30 independent runs. The results of all PSO variants and different variants of SSG-PSO with different local search techniques are gained from [50].

**Table 8** Mean and standard deviation results obtained by SGO compared with SAABC, GABC, IABC, ABC/Best1, GPSO, DBMPSO, TCPSO and VABC on the multidimensional benchmark functions after 30 independent runs for  $\text{dim} = 60$ 

Function	SAABC	GABC	IABC	ABC/Best1	CPSO	DBMPSO	TCPSO	VABC	SGO
$F_1$	6.74e-02 (9.67e-03)-	1.52e-01 (1.42e-01)-	2.58e-07 (1.63e-07)-	1.86e+01 (9.13e+00)-	1.58e+01 (4.38e+00)-	1.19e+01 (3.18e+00)-	4.01e+00 (9.18e-01)-	1.21e-28 (1.67e-28)-	0
$F_2$	3.59e+04 (3.23e+04)-	5.99e+03 (9.61e+03)-	3.83e+01 (8.56e+01)-	1.03e+05 (6.99e+04)-	1.05e+05 (2.86e+04)-	1.03e+05 (2.33e+04)-	5.02e+04 (1.60e+04)-	8.73e-23 (1.95e-22)-	0
$F_3$	2.38e+00 (1.95e+00)-	2.53e+00 (1.61e+00)-	3.62e-02 (7.69e-02)-	1.22e+01 (2.31e+00)-	4.76e+01 (3.48e+00)-	4.11e+01 (6.17e+00)-	2.15e+01 (6.42e+00)-	5.49e-19 (1.23e-18)-	0
$F_4$	6.26e+01 (4.04e+00)-	9.22e+01 (1.22e+00)-	9.19e+01 (2.07e+00)-	9.16e+01 (3.84e+00)-	2.38e+01 (1.49e+00)-	2.59e+01 (3.29e+00)-	1.82e+01 (2.18e+00)-	7.02e-06 (9.88e-05)-	0
$F_5$	1.81e+05 (1.99e+04)-	1.76e+05 (2.98e+04)-	1.48e+05 (4.78e+04)-	2.01e+05 (2.75e+04)-	1.35e+04 (2.87e+03)-	7.58e+03 (1.84e+03)-	5.16e+03 (1.84e+03)-	1.19e-37 (2.67e-37)-	0
$F_6$	4.05e+05 (3.13e+05)-	1.55e+04 (1.01e+04)-	2.07e+03 (1.79e+03)-	4.16e+05 (3.13e+05)-	1.07e+06 (4.98e+05)-	4.52e+05 (2.08e+05)-	3.45e+05 (3.32e+05)-	1.90e-13 (1.70e-13)+	54.4397 1.0082
$F_7$	1.61e+03 (9.87e+01)-	1.72e+03 (1.50e+02)-	1.58e+03 (1.64e+02)-	1.70e+03 (8.99e+01)-	2.25e+02 (6.09e+01)-	1.36e+02 (2.57e+01)-	1.03e+02 (4.16e+01)-	8.25e-28 (1.40e-27)+	0.6667 2.09e-05
$F_8$	0 (0)≈	2.64e+03 (6.81e+02)-	1.20e+03 (5.33e+02)-	6.82e+03 (3.21e+03)-	4.80e+03 (6.35e+02)-	4.85e+03 (1.78e+03)-	2.10e+03 (7.01e+02)-	0 (0)≈	0
$F_9$	1.56e+00 (3.27e-01)-	7.89e-01 (7.63e-02)-	1.22e-02 (5.16e-03)-	7.27e-01 (8.09e-02)-	2.10e+00 (3.93e-01)-	1.48e+00 (3.01e-01)-	1.06e+00 (3.58e-01)-	6.97e-03 (7.58e-03)-	1.68e-05 1.62e-05
$F_{10}$	1.11e+04 (1.15e+04)-	6.37e+02 (1.23e+03)-	5.44e+00 (4.43e+00)-	1.78e+05 (3.78e+05)-	5.41e+06 (1.74e+06)-	3.08e+06 (1.56e+06)-	1.79e+06 (1.33e+06)-	9.72e-01 (2.40e-09)-	0
$F_{11}$	3.29e+02 (5.09e+01)-	2.47e+02 (6.76e+01)-	1.81e+02 (2.88e+01)-	2.66e+02 (5.10e+01)-	3.42e+02 (2.48e+01)-	3.66e+02 (6.41e+01)-	2.08e+02 (1.82e+01)-	0 (0)≈	0
$F_{12}$	1.14e+01 (1.74e+00)-	9.50e+00 (6.26e-01)-	7.33e+00 (1.12e+00)-	1.02e+01 (2.14e+00)-	1.12e+01 (7.97e-01)-	1.21e+01 (6.94e-01)-	9.26e+00 (5.14e-01)-	5.84e-11 (8.76e-11)-	-8.89e-16 0
$F_{13}$	5.05e+01 (8.14e+01)-	-1.27e+01 (1.40e+01)-	-3.88e+01 (1.27e+01)-	7.76e+00 (1.99e+01)-	5.60e+01 (9.79e+00)-	5.59e+01 (1.76e+01)-	1.51e+01 (1.02e+01)-	-60 (0)-	0
$F_{14}$	7.20e+06 (7.28e+06)-	5.34e+05 (8.71e+05)-	3.52e+05 (6.80e+05)-	6.43e+04 (4.58e+04)-	2.15e+02 (2.96e+02)-	5.70e+02 (2.30e+02)-	1.02e+03 (1.07e+02)-	3.41e+03 (2.07e-04)-	1.65e-06 5.65e-06
$F_{15}$	2.52e+07 (3.00e+07)-	1.95e+06 (2.97e+06)-	2.52e+05 (3.11e+05)-	9.33e+06 (1.86e+07)-	2.24e+05 (2.14e+05)-	6.50e+04 (4.30e+04)-	1.86e+03 (3.11e+03)-	1.87e+03 (5.57e-05)-	3.14e-05 2.11e-05
-	14	15	15	15	15	15	15	11	
+	00	00	00	00	00	00	00	02	
≈	1	00	00	00	00	00	00	02	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each, SAABC, GABC, IABC, ABC/Best1, CPSO, DBMPSO, TCPSO and VABC. "−", "+", and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

**Table 9** Mean and standard deviation results obtained by SGO comparing with SAABC, GABC, IABC, ABC/Best1, GPSO, DBMPSO, TCPSO and VABC on the multidimensional benchmark functions after 30 independent runs for dim = 100

Function	SAABC	GABC	IABC	ABC/Best1	CPSO	DBMPSO	TCPSO	VABC	SGO
$F_1$	1.64e+00 (6.61e-01)-	4.45e+00 (1.79e+00)-	3.52e-03 (4.74e-03)-	4.54e+01 (2.37e+01)-	2.74e+01 (3.79e+00)-	2.13e+01 (1.58e+00)-	1.22e+01 (1.45e+00)-	1.05e-25 (2.34e-25)-	0
$F_2$	9.96e+05 (7.98e+05)-	1.44e+05 (8.29e+04)-	1.24e+01 (1.02e+01)-	1.46e+06 (5.47e+05)-	5.57e+05 (1.06e+05)-	3.58e+00 (9.14e+04)-	2.13e+05 (4.70e+04)-	3.67e-31 (8.20e-31)-	0
$F_3$	2.48e+01 (7.37e+00)-	1.02e+01 (2.29e+00)-	1.44e-01 (7.90e-02)-	3.66e+01 (1.38e+01)-	8.88e+01 (5.51e+00)-	7.94e+01 (5.13e+00)-	5.87e+01 (5.53e+00)-	2.13e-12 (4.77e-12)-	0
$F_4$	7.99e+01 (3.51e+00)-	9.61e+01 (6.72e-01)-	9.48e+01 (1.03e+01)-	9.57e+01 (2.77e-01)-	2.70e+01 (2.29e+00)-	2.60e+01 (1.87e+00)-	2.41e+01 (2.97e+00)-	1.59e-04 (2.17e-04)-	0
$F_5$	4.43e+05 (7.35e+04)-	5.03e+05 (7.35e+04)-	3.91e+05 (1.21e+05)-	5.43e+05 (7.95e+04)-	4.20e+04 (1.32e+04)-	2.29e+04 (5.51e+03)-	1.48e+04 (2.44e+03)-	1.36e-33 (3.04e-33)-	0
$F_6$	2.00e+07 (2.17e+07)-	7.59e+05 (9.43e+05)-	1.25e+05 (1.66e+05)-	1.19e+07 (6.16e+06)-	3.31e+06 (1.06e+06)-	1.38e+06 (3.53e+05)-	5.09e+05 (1.10e+05)-	1.60e-11 (2.62e-11)+	94.7977
$F_7$	3.01e+03 (2.33e+02)-	3.05e+03 (2.53e+02)-	2.84e+03 (1.49e+02)-	3.10e+03 (2.53e+02)-	5.01e+02 (7.71e+01)-	3.36e+02 (5.15e+01)-	3.06e+02 (7.24e+01)-	7.75e-31 (1.73e-30)+	1.0316
$F_8$	3.22e+02 (1.95e+02)-	1.47e+04 (3.51e+03)-	5.27e+03 (2.87e+03)-	19.936 (4.98e+03)-	1.16e+04 (1.34e+03)-	8.88e+03 (8.22e+02)-	5.25e+03 (9.66e+02)-	0 (0)≈	0
$F_9$	5.30e+00 (9.95e-01)-	2.18e+00 (2.21e-01)-	9.51e-02 (3.37e-02)-	2.68e+00 (1.89e-01)-	5.91e+00 (1.42e+00)-	3.85e+00 (7.73e-01)-	1.86e+00 (4.94e-01)-	1.02e-02 (9.49e-03)-	1.49e-05
$F_{10}$	2.29e+07 (1.67e+07)-	3.40e+06 (7.38e+06)-	2.65e+04 (5.93e+04)-	9.85e+06 (9.94e+06)-	4.47e+07 (1.47e+07)-	3.55e+07 (4.51e+06)-	1.45e+07 (4.32e+06)-	9.83e-01 (4.20e-08)-	0
$F_{11}$	6.84e+02 (6.80e+01)-	6.08e+02 (1.07e+02)-	4.72e+02 (4.20e+01)-	7.50e+02 (7.88e+01)-	6.86e+02 (3.22e+01)-	6.41e+02 (2.53e+01)-	5.11e+02 (1.68e+01)-	3.34e-14 (7.47e-14)-	0
$F_{12}$	1.35e+01 (1.63e+00)-	1.27e+01 (1.64e+00)-	1.13e+01 (1.41e+00)-	1.55e+01 (8.21e-01)-	1.26e+01 (8.02e-01)-	1.21e+01 (6.24e-01)-	9.28e+00 (5.94e-01)-	1.50e-05 (3.33e-05)-	-8.89e-16
$F_{13}$	2.44e+02 (7.50e+01)-	6.11e+01 (6.08e+01)-	4.02e+01 (2.67e+01)-	1.40e+02 (5.38e+01)-	1.23e+02 (2.66e+01)-	1.25e+02 (3.07e+01)-	7.00e+01 (1.91e+01)-	-100 (0)-	0
$F_{14}$	3.28e+07 (3.49e+07)-	2.33e+07 (2.16e+07)-	1.18e+07 (1.08e+07)-	2.13e+07 (1.91e+07)-	4.90e+04 (4.93e+04)-	1.85e+02 (1.19e+03)-	9.51e+02 (2.47e+02)-	5.72e+03 (4.35e-06)-	2.06e-05
$F_{15}$	9.93e+07 (9.03e+07)-	4.53e+07 (5.05e+07)-	5.97e+06 (5.07e+06)-	4.56e+07 (3.07e+07)-	1.07e+06 (7.55e+05)-	5.59e+05 (2.36e+05)-	9.49e+03 (5.76e+03)-	3.14e+03 (3.19e-06)-	3.62e-05
-	15	15	15	15	15	15	15	12	1.15e-04
+	00	00	00	00	00	00	00	02	2.23e-04
≈	00	00	00	00	00	00	00	01	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each, SAABC, GABC, IABC, ABC/Best1, CPSO, DBMPSO, TCPSO and VABC. "−", "+", and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively



**Table 10** Mean and standard deviation results obtained by SGO comparing with SAABC, GABC, IABC, ABC/Best1, GPSO, DBMPSO, TCPSO and VABC on the fixed-dimensional benchmark functions after 30 independent runs

Function	SAABC	GABC	IABC	ABC/Best1	CPSO	DBMPSO	TCPSO	VABC	SGO
$F_{16}$	9.20e-06 (0.55e-06)–	3.42e-05 (4.79e-05)–	4.77e-02 (1.66e-02)–	7.09e-08 (1.58e-07)–	0 (0)≈	0 (1)≈	0 (2)≈	7.17e-04 (8.76e-04)–	0 0
$F_{17}$	1.16e-05 (1.07e-05)–	3.38e-05 (2.83e-05)–	6.00e-02 (4.75e-02)–	5.95e-07 (1.08e-06)–	0 (0)≈	0 (1)≈	0 (2)≈	1.68e-03 (1.35e-03)–	0 0
$F_{18}$	2.68e-01 (6.73e-05)≈	2.68e-01 (8.37e-14)≈	2.69e-01 (1.14e-03)≈	2.68e-01 (1.24e-16)≈	2.68e-01 (0)≈	2.68e-01 (0)≈	2.68e-01 (0)≈	2.95e-01 (2.85e-02)–	2.68e-01 2.26e-16
$F_{19}$	1.80e-02 (3.51e-02)–	8.83e-02 (5.78e-02)–	1.63e+01 (1.95e+00)–	8.77e-02 (1.96e-01)–	–1.14e-13 (0)–	–6.82e-14 (4.76e-14)≈	–1.14e-130 (0)–	1.96e+01 (3.34e-01)–	–1.16e-14 6.06e-14
$F_{20}$	1.06e+02 (7.08e+01)–	3.52e+01 (2.21e+01)–	1.31e+02 (2.11e+01)–	1.58e+01 (8.86e+00)–	3.18e+01 (3.30e+01)–	1.13e+01 (1.92e+01)–	–1.82e-13 (1.19e-12)+	1.20e+02 (1.50e-02)–	3.16e-08 1.73e-07
$F_{21}$	0 (0)≈	–4.78e-01 (1.73e-02)–	–4.83e-01 (1.90e-02)–	–4.95e-01 (4.70e-03)–	–4.90e-01 (6.21e-17)–	–4.98e-01 (4.35e-03)–	–5.00e-01 (0)–	–5.00e-01 (0)–	0 0
$F_{22}$	3.00e+00 (8.18e-05)≈	3.00e+00 (3.13e-12)≈	3.23e+00 (4.50e-01)–	3.00e+00 (5.50e-15)≈	3.00e+00 (1.06e-15)≈	3.00e+00 (4.97e-16)≈	3.00e+00 (1.02e-15)≈	3.26e+01 (2.93e-02)–	3.00 2.82e-15
$F_{23}$	–1.97e-01 (8.02e-02)+	–1.78e-07 (1.43e-09)+	1.61e+00 (4.21e-01)–	2.97e-05 (4.61e-05)+	4.43e-01 (3.31e-01)+	1.70e-01 (7.42e-02)+	1.86e+00 (2.83e-01)–	2.11e-01 (2.94e-01)+	1.23 6.65e-04
–	04	05	07	05	03	02	03	07	
+	01	01	00	01	01	01	01	01	
≈	03	02	01	02	04	05	04	00	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each, SAABC, GABC, IABC, ABC/Best1, CPSO, DBMPSO, TCPSO and VABC. “–”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

**Table 11** Experimental results of GPSO, LPSO, FPS, SPSP, CLPSO, OLPSO, SLPSOA and SGO over 25 independent runs on 14 test functions of 30 variables with 200,000 FEs

Functions	GPSO	LPSO	FPS	SPSO	CLPSO	OLPSO	SLPSOA	SGO
Sphere	2.05e-32 (3.56e-32)-	3.34e-14 (5.39e-14)-	2.42e-13 (1.73e-13)-	2.29e-96 (9.48e-96)-	1.58e-12 (7.70e-13)-	1.11e-38 (1.28e-38)-	7.30e-38 (8.14e-38)-	0.00e+00 0.00e+00
Schwefel 2.22	1.49e-21 (3.60e-21)-	1.70e-10 (1.39e-10)-	2.76e-08 (9.04e-09)-	1.74e-53 (1.58e-53)-	2.51e-08 (5.84e-09)-	7.67e-22 (56.3e-22)-	0.00e+00 (0.00e+00)≈	0.00e+00 0.00e+00
Rosenbrock	40.70 (32.19)-	28.08 (21.79)-	25.12 (0.51)-	13.50 (14.63)+	11.36 (9.85)+	1.26 (1.40)+	0.46 (0.52)+	18.31 0.92
Noise	9.32e-03 (2.39e-03)-	2.28e-02 (5.60e-03)-	4.24e-03 (1.28e-03)-	4.02e-03 (1.66e-03)-	5.85e-03 (1.11e-03)-	1.64e-02 (3.25e-03)-	3.14e-03 (1.02e-03)-	1.02e-06 1.62e-05
Schwefel	2.48e+03 (2.97e+02)+	3.16e+03 (4.06e+02)-	9.93e+02 (5.09e+02)+	3.14e+03 (7.81e+02)+	3.82e-04 (1.28e-04)+	3.82e-04 (0.00e+00)+	15.4 (3.58)+	3.15e+03 779.1223
Rastrigin	26.03 (7.27)-	35.07 (6.89)-	65.10 (13.39)-	41.03 (11.09)-	9.09e-05 (1.25e-04)-	0.00e+00 (0.00e+00)≈	2.09e-05 (3.20e-05)-	0.00e+00 0.00e+00
Ackley	1.31e-14 (2.08e-15)-	8.20e-08 (6.73e-08)-	2.33e-07 (7.19e-08)-	3.73e-02 (0.19)-	3.66e-07 (7.57e-08)-	4.14e-15 (0.00e+00)-	2.24e-16 (5.02e-17)-	-8.88e-16 0.00e+00
Griewank	2.12e-02 (2.18e-02)-	1.53e-03 (4.32e-03)-	9.01e-12 (1.84e-11)-	7.48e-03 (1.25e-02)-	9.02e-10 (8.57e-09)-	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	0.00e+00 0.00e+00
Generalized penalized	2.23e-31 (7.07e-31)-	8.10e-16 (1.07e-15)-	1.96e-15 (1.11e-15)-	7.47e-02 (3.11)-	6.45e-14 (3.70e-14)-	1.57e-32 (2.79e-48)≈	3.32e-35 (7.76e-35)+	1.78e-32 3.01e-33
Generalized penalized I	1.32e-03 (3.64e-03)-	3.26e-13 (3.70e-13)-	2.70e-14 (1.57e-14)-	1.76e-03 (4.11e-03)-	1.25e-12 (9.45e-13)-	1.35e-32 (5.59e-48)+	4.39e-34 (2.20e-35)+	8.62e-22 2.43e-21
Rotated schwefel	4.61e+03 (6.21e+02)-	4.50e+03 (3.97e+02)-	4.41e+03 (9.94e+02)-	4.57e+03 (6.28e+02)-	4.39e+03 (3.51e+03)-	3.13e+03 (1.24e+03)+	2.72e+03 (5.95e+02)+	4.00e+03 3.24e+02
Rotated rastrigin	60.02 (15.98)-	53.36 (13.99)-	1.50e+02 (14.48)-	43.42 (17.38)-	87.14 (10.76)-	53.35 (13.35)-	40.59 (12.46)-	0.00e+00 0.00e+00
Rotated Ackley	1.93 (0.96)-	1.55 (0.45)-	3.16e-07 (1.00e-07)-	9.24e-02 (0.32)-	5.91e-05 (6.46e-05)-	4.28e-15 (7.11e-16)≈	2.07e-14 (6.68e-15)-	-8.88e-16 0.00e+00
Rotated griewank	1.80e-02 (2.41e-02)-	1.68e-03 (3.47e-03)-	1.28e-08 (4.29e-08)-	3.05e-03 (5.70e-03)-	7.96e-05 (7.66e-05)-	4.19e-08 (2.06e-07)-	6.50e-09 (2.31e-08)-	0.00e+00 0.00e+00
-	13	14	13	12	12	06	07	
+	01	00	01	02	02	04	05	
≈	00	00	00	00	00	04	02	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each GPSO, LPSO, FPS, SPSP, CLPSO, OLPSO and SLPSOA. “-”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

**Table 12** Experimental results of FIPS-PSO, CPSO-H, DMSPSO-LS, CLPSO APSO, SSG-PSO, SSG-PSO-DFP, SSG-PSO-BFGS, SSG-PSO-NM, SSG-PSO-PS, and SGO over 30 independent runs on 21 test functions of 30 variables with 300,000 FEs

Algorithm	Sphere Mean Std	Rosenbrock Mean Std	Schwefel 12 Mean Std	Schwefel 2.21 Mean Std	Schwefel 2.22 Mean Std	Rastrigin Mean Std	Noncontinuous rastrigin Mean Std	–/+ /≈
FIPS-PSO	0.00e+00 (0.00e+00)≈	2.52e+01 (9.08–01)–	2.08e+02 (8.98e+01)–	6.25e–02 (4.34e–03)–	1.64e–09 (5.59e–10)–	6.39e+01 (1.12e+01)–	5.44e+01 (2.63e+01)–	06/00/01
CPSO-H	2.49e–15 (1.55e–15)–	2.77e+01 (2.86e+01)–	2.79e+03 (5.98e+03)–	5.14e–03 (4.49e–03)–	3.92e–08 (4.25e–09)–	3.32e–02 (1.82e–01)–	2.56e–09 (7.43e–09)–	07/00/00
CLPSO	0.00e+00 (0.00e+00)≈	2.34e+01 (3.79e+00)–	1.34e+02 (2.44e+02)–	6.61e–03 (4.21e–03)–	7.52e–20 (5.69e–19)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	04/00/03
APSO	0.00e+00 (0.00e+00)≈	2.38e+01 (7.05e+01)–	9.91e–03 (5.03e–02)–	2.24e–01 (6.85e–01)–	4.25e–23 (7.44e–21)–	4.52e+00 (1.35e+00)–	3.21e+00 (6.32e+00)–	06/00/01
DMSPSO-LS	0.00e+00 (0.00e+00)≈	8.99e–11 (3.54e–11)+	9.59e–10 (5.33e–10)–	4.67e–05 (1.75e–05)–	1.09e–18 (1.04e–18)–	1.32e+01 (1.93e+00)–	3.41e+01 (5.02e+00)–	05/01/01
SSG-PSO	0.00e+00 (0.00e+00)≈	2.12e+01 (1.84e+00)–	1.24e+01 (1.35e+01)–	2.77e–02 (1.47e–02)–	2.75e–25 (1.96e–25)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	04/00/03
SSG-PSO-DFP	0.00e+00 (0.00e+00)≈	1.84e–01 (4.35e–01)+	1.30e–03 (2.12e–03)–	1.60e–04 (2.92e–06)–	9.65e–25 (6.08e–25)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	03/01/03
SSG-PSO-BFGS	0.00e+00 (0.00e+00)≈	5.73e–11 (8.08e–12)+	2.57e–14 (4.58e–14)–	3.77e–06 (1.63e–06)–	1.04e–26 (7.54e–26)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	03/01/03
SSG-PSO-NM	0.00e+00 (0.00e+00)≈	8.34e+00 (2.83e+00)+	2.46e–02 (4.14e–02)–	2.12e–02 (1.11e–02)–	1.93e–24 (1.21e–24)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	03/01/03
SSG-PSO-PS	0.00e+00 (0.00e+00)≈	6.90e+00 (1.25e+00)+	4.16e+01 (2.31e+00)–	1.23e–15 (3.25e–16)–	9.33e–22 (1.23e–22)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	03/01/03
SGO	0.00e+00 0.00e+00	1.52e+01 1.31e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	
	Ackley	Griewank	Scaled Rosenbrock 100	Scaled rastrigin 10	Scaled rastrigin 1000	Rotated sphere	Rotated Rosenbrock	
FIPS-PSO	1.39e–08 (2.98e–09)–	2.72e–07 (1.18e–06)–	7.37e+04 (3.14e+05)–	7.37e+04 (9.23e+00)–	4.59e+01 (2.38e+01)–	7.54e–13 (3.26e–13)–	2.89e+01 (4.15e+00)–	07/00/00
CPSO-H	2.44e–05 (1.35e–05)–	1.20e–01 (2.18e–01)–	3.71+06 (4.38e+06)–	1.23e+07 (1.86e–07)–	5.65e–05 (2.44e–04)–	8.10e–08 (1.02e–07)–	1.62e+02 (3.78e+02)–	07/00/00
CLPSO	7.77e–14 (1.49e–18)–	0.00e+00 (0.00e+00)≈	4.94e+01 (4.32e+01)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	4.21e–17 (6.18e–18)–	2.64e+01 (1.21e+00)+	03/01/03
APSO	6.34e–02 (1.43e+00)–	1.42e–02 (7.24e–02)–	4.06e+06 (5.42e+06)–	1.98e+00 (2.44e+01)–	1.49e+01 (5.24e+01)–	3.24e–20 (5.45e–19)–	7.83e+01 (8.24e+01)–	07/00/00
DMSPSO-LS	7.81e–15 (2.80e–15)–	0.00e+00 (0.00e+00)≈	2.56e+01 (1.02e+01)–	2.17e+01 (6.70e+00)–	3.06e+01 (6.41e+00)–	2.59e–30 (1.87e–30)–	3.98e–03 (2.35e–03)+	05/01/01
SSG-PSO	7.25e–15 (1.74e–16)–	0.00e+00 (0.00e+00)≈	3.59e+01 (3.22e+01)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	5.28e–22 (8.71e–22)–	2.53e+01 (6.91e–01)+	03/01/03
SSG-PSO-DFP	5.68e–15 (1.78e–15)–	0.00e+00 (0.00e+00)≈	3.19e+01 (3.18e+01)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	1.91e–23 (3.04e–23)–	1.06e–05 (7.88e–04)+	03/01/03
SSG-PSO-BFGS	4.97e–15 (1.73e–15)–	0.00e+00 (0.00e+00)≈	2.21e+01 (3.07e+01)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	2.29e–27 (3.74e–27)–	3.98e–10 (1.22e–10)+	03/01/03
SSG-PSO-NM	6.85e–15 (3.23e–15)–	0.00e+00 (0.00e+00)≈	3.43e+01 (3.03e+01)–	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	5.32e–24 (4.11e–24)–	2.40e+01 (1.36e+01)+	03/01/03
SSG-PSO-PS	1.25e–14 (6.32e–15)–	0.00e+00 (0.00e+00)≈	1.81e+00 (1.05e+00)+	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	1.97e–24 (8.65e–25)–	2.50e+01 (1.41e+01)+	02/02/03
SGO	–8.88e–16 0.00e+00	0.00e+00 0.00e+00	1.68e+01 2.45e–01	0.00e+00 0.00e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	2.82e+01 1.26e–02	

**Table 12** continued

Algorithm	Sphere Mean Std	Rosenbrock Mean Std	Schwefel 12 Mean Std	Schwefel 2.21 Mean Std	Schwefel 2.22 Mean Std	Rastrigin Mean Std	Noncontinuous rastrigin Mean Std	–/+ /≈
	Rotated schwefel 2.21	Rotated rastrigin	Rotated Ackley	Rotated griewank	Rotated ellipse	Rotated tablet	Rotated diff pow	
FIPS-PSO	1.36e–04 (4.89e–05)–	1.75e+02 (8.79e+00)–	2.24e–08 (5.60e–09)–	1.14e–03 (3.00e–03)–	1.51e+03 (7.14e+02)–	8.45e+02 (2.14e+02)–	4.54e+10 (9.01e+10)–	07/00/00
CPSO-H	5.43e+01 (7.52e+00)–	3.77e+02 (1.10e+02)–	1.76e+01 (3.96e+00)–	1.66e+00 (2.10e–01)–	7.63e+03 (6.69e+03)–	1.65e+05 (4.61e+04)–	1.6e+10 (4.71e+10)–	07/00/00
CLPSO	1.51e–02 (4.64e–03)–	1.08e+02 (1.36e+01)–	2.76e–03 (3.25e–03)–	2.66e–03 (2.12e–03)–	4.88e+03 (1.38e+03)–	3.35e+02 (1.34e+02)–	3.74e+07 (3.45e+07)–	07/00/00
APSO	8.05e–01 (1.24e–01)–	1.02e+02 (1.24e+03)–	3.62e–10 (9.94e–10)–	1.72e–02 (2.41e–01)–	1.25e+03 (2.12e+04)–	7.41e+02 (8.44e+02)–	2.91e+07 (4.22e+08)–	07/00/00
DMSPSO-LS	1.18e–04 (1.09e–04)–	3.10e+01 (4.54e+00)–	2.48e–14 (5.84e–15)–	7.39e–04 (2.33e–03)–	1.27e–09 (1.13e–09)–	1.37e–07 (1.47e–07)–	1.28e–08 (6.28e–09)–	07/00/00
SSG-PSO	7.25e–04 (3.02e–04)–	4.65e+01 (1.12e+01)–	5.86e–14 (1.55e–13)–	1.02e–05 (4.86e–05)–	7.72e+02 (8.86e+02)–	3.02e+02 (1.10e+02)–	5.42e+06 (7.17e+06)–	07/00/00
SSG-PSO-DFP	5.38e–06 (8.59e–06)–	5.08e+01 (1.16e+01)–	4.79e–15 (1.73e–15)–	1.11e–16 (3.02e–16)–	2.54e–07 (1.07e–06)–	1.30e–11 (4.44e–11)–	1.45e–06 (9.86e–07)–	07/00/00
SSG-PSO-BFGS	8.15e–06 (7.15e–06)–	4.10e+01 (1.43e+01)–	4.59e–15 (2.64e–15)–	1.47e–16 (3.03e–16)–	2.44e–16 (4.38e–16)–	2.23e–12 (2.40e–12)–	8.87e–10 (5.79e–10)–	07/00/00
SSG-PSO-NM	4.94e–04 (1.96e–04)–	5.38e+00 (1.05e+01)–	7.12e–14 (1.67e–13)–	1.10e–08 (2.70e–08)–	6.17e–01 (8.93e–01)–	4.91e–11 (2.10e–10)–	7.73e+03 (1.44e+04)–	07/00/00
SSG-PSO-PS	4.40e–04 (2.31e–04)–	4.44e+01 (6.43e+01)–	6.16e–14 (1.95e–13)–	1.52e–06 (2.84e–06)–	5.13e+01 (3.24e+01)–	2.14e–02 (6.54e–02)–	3.26e+04 (2.16e+04)–	07/00/00
SGO	0.00e+00 0.00e+00	0.00e+00 0.00e+00	–8.88e–16 0.00e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	0.00e+00 0.00e+00	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each FIPS-PSO, CPSO-H, CLPSO, APSO, DMSPSO-LS, SSG-PSO, SSG-PSO-DFP, SSG-PSO-BFGS, SSG-PSO-NM, and SSG-PSO-PS. “–”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

As seen from Table 12 results, SGO found the global optimal solution for all the test functions except Rosenbrock, Ackley, scaled Rosenbrock 100, rotated Rosenbrock and rotated Ackley. On the other hand, for the test functions Ackley and rotated Ackley, the objective values obtained by SGO are extremely close to global optima. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Table summarize the results. According to Wilcoxon's rank-sum test, SGO performs superior than FIPS-PSO, CPSO-H, DMSPSO-LS, CLPSO, APSO, SSG-PSO, SSG-PSO-DFP, SSG-PSO-BFGS, SSG-PSO-NM, and SSG-PSO-PS in 20, 21, 17, 14, 20, 14, 13, 13, 13 and 12 test functions, respectively, and equivalent with 1, 0, 2, 6, 1, 6, 6, 6 and 6 test functions, respectively, out of 21 test functions. So, it is interesting to tell according to this experiment that SGO is better than other algorithms according to Wilcoxon's rank-sum test.

### Experiment 9: SGO vs. jDE, SaDE, EPSDE, CoDE, MPEDE, CLPSO, CMA-ES, GL-25 and TLBO

To study the performance of proposed SGO, 25 test functions proposed in the 2005 special session on real parameter optimization were used. A detailed description of these test functions can be found in [21]. The number of decision variables or dimension of the function was set to 30 for all test functions. For each algorithm and each test function, 25 independent runs were conducted with 300,000 function evaluations (FEs) as the termination.

SGO was compared with six DE variants, i.e., JADE [32], jDE [31], SaDE [30], EPSDE [34], CoDE [33] and MPEDE [51] and four other approaches, i.e., CLPSO [28], CMA-ES [52], GL-25 [53] and TLBO [24]. In our experiments, the parameter settings of these methods were the same as their original papers. The number of FFs in all these methods was 300,000, and each method was run 25 times on each test function. For the proposed SGO algorithm, we

**Table 13** Experimental results of JADE, jDE, SaDE, EPSDE, CoDE, MPEDE and SGO over 25 independent runs on 25 test functions of 30 variables with 300,000 FEs

Function	JADE Mean error Std	jDE Mean error Std	SaDE Mean error Std	EPSDE Mean error Std	CoDE Mean error Std	MPEDE Mean error Std	SGO Mean error Std
$F_1$	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)≈	0.00e+00 (0.00e+00)
$F_2$	1.07e−28 (1.00e−28)+	1.11e−06 (1.10e−06)−	8.26e−06 (1.65e−05)−	4.23e−26 (4.07e−26)+	1.69e−15 (3.95e−15)+	1.01e−26 (2.05e−26)+	6.79e−09 (4.79e−09)
$F_3$	8.42e+03 (7.26e+03)+	1.98e+05 (1.10e+05)−	4.27e+05 (2.08e+05)−	8.74e+05 (3.28e+06)−	1.05e+05 (6.25e+04)+	1.01e+01 (8.32e+00)+	1.65e+05 (9.07e+04)
$F_4$	1.73e−16 (5.43e−16)+	4.40e−02 (1.26e−01)+	1.77e+02 (2.67e+02)−	3.49e+02 (2.23e+03)−	5.81e−03 (1.38e−02)+	6.61e−16 (5.68e−16)+	2.06e+01 (1.26e+01)
$F_5$	8.59e−08 (5.23e−07)+	5.11e+02 (4.40e+02)+	3.25e+03 (5.90e+02)−	1.40e+03 (7.12e+02)+	3.31e+02 (3.44e+02)+	7.21e−06 5.12e−06+	2.08e+03 (4.81e+02)
$F_6$	1.02e+01 (2.96e+01)−	2.35e+01 (2.50e+01)−	5.31e+01 (3.25e+01)−	6.38e−01 (1.49e+00)+	1.60e−01 (7.85e−01)+	9.65e+00 (4.65e+00)−	2.63e+00 (1.74e+00)
$F_7$	8.07e−03 (7.42e−03)−	1.18e−02 (7.78e−03)−	1.57e−02 (1.38e−02)−	1.77e−02 (1.34e−02)−	7.46e−03 (8.55e−03)−	2.36e−03 (1.15e−03)+	7.00e−03 (5.60e−03)
$F_8$	2.09e+01 (1.68e−01)−	2.09e+01 (4.86e−02)−	2.09e+01 (4.95e−02)−	2.09e+01 (5.81e−02)−	2.01e+01 (1.41e−01)+	2.09e+01 (5.87e−01)−	2.08e+01 (5.83e−03)
$F_9$	0.00e+00 (0.00e+00)+	0.00e+00 (0.00e+00)+	2.39e−01 (4.33e−01)+	3.98e−02 (1.99e−01)+	0.00e+00 (0.00e+00)+	0.00e+00 (0.00e+00)+	9.27e+01 (2.15e+01)
$F_{10}$	2.41e+01 (4.61e+00)+	5.54e+01 (8.46e+00)+	4.72e+01 (1.01e+01)+	5.36e+01 (3.03e+01)+	4.15e+01 (1.16e+01)+	1.52e+01 (2.98e+00)+	9.60e+01 (1.18e+01)
$F_{11}$	2.53e+01 (1.65e+00)−	2.79e+01 (1.61e+00)−	1.65e+01 (2.42e+00)−	3.56e+01 (3.88e+00)−	2.71e+01 (1.57e+00)−	2.58e+01 (3.11e+00)−	1.72e+01 (1.54e+00)
$F_{12}$	6.15e+03 (4.79e+03)−	8.63e+03 (8.31e+03)−	3.02e+03 (2.33e+03)−	3.58e+04 (7.05e+03)−	3.05e+03 (3.80e+03)−	1.17e+03 (8.66e+02)−	4.50e+02 (2.12e+01)
$F_{13}$	1.49e+00 (1.09e−01)+	1.66e+00 (1.35e−01)+	3.94e+00 (2.81e−01)−	1.94e+00 (1.46e−01)+	1.57e+00 (3.27e−01)+	2.92e+00 (6.33e−01)+	3.26e+00 (4.57e−01)
$F_{14}$	1.23e+01 (3.11e−01)−	1.30e+01 (2.00e−01)−	1.26e+01 (2.83e−01)−	1.35e+01 (2.09e−01)−	1.23e+01 (4.81e−01)−	1.23e+01 (4.22e−01)−	1.17e+01 (3.29e−01)
$F_{15}$	3.51e+02 (1.28e+02)−	3.77e+02 (8.02e+01)−	3.76e+02 (7.83e+01)−	2.12e+02 (1.98e+01)+	3.88e+02 (6.85e+01)−	3.78e+02 (6.32e+01)−	2.75e+02 (6.56e+01)
$F_{16}$	1.01e+02 (1.24e+02)+	7.94e+01 (2.96e+01)+	8.57e+01 (6.94e+01)+	1.22e+02 (9.19e+01)−	7.37e+01 (5.13e+01)+	3.77e+01 (5.22e+00)+	1.12e+02 (6.22e+01)
$F_{17}$	1.47e+02 (1.33e+02)−	1.37e+02 (3.80e+01)+	7.83e+01 (3.76e+01)+	1.69e+02 (1.02e+02)−	6.67e+01 (2.12e+01)+	4.36e+01 (6.35e+00)+	1.62e+02 (5.91e+00)
$F_{18}$	9.04e+02 (1.03e+00)−	9.04e+02 (1.08e+01)−	8.68e+02 (6.23e+01)+	8.20e+02 (3.35e+00)+	9.04e+02 (1.04e+00)−	9.04e+02 (1.21e+00)−	9.00e+02 (0.00e+00)
$F_{19}$	9.04e+02 8.40e+01	9.04e+02 1.11e+00	8.74e+02 6.22e+01	8.21e+02 (3.35e+00)+	9.04e+02 (9.42e−01)−	9.04e+02 (1.24e+00)−	9.00e+02 (0.00e+00)
$F_{20}$	9.04e+02 (8.47e−01)−	9.04e+02 (1.10e+00)−	8.78e+02 (6.03e+01)+	8.22e+02 (4.17e+00)+	9.04e+02 (9.01e−01)−	9.04e+02 (1.18e+00)−	9.00e+02 (0.00e+00)
$F_{21}$	5.00e+02 (4.67e−13)−	5.00e+02 (4.80e−13)−	5.52e+02 (1.82e+02)−	8.33e+02 (1.00e+02)−	5.00e+02 (4.88e−13)−	5.00e+02 (3.54e−14)−	4.79e+02 (2.03e+01)
$F_{22}$	8.66e+02 (1.91e+01)−	8.75e+02 (1.91e+01)−	9.36e+02 (1.83e+01)−	5.07e+02 (7.26e+00)−	8.63e+02 (2.43e+01)−	8.72e+02 (2.98e+01)−	4.68e+02 (2.12e+01)

**Table 13** continued

Function	JADE Mean error Std	jDE Mean error Std	SaDE Mean error Std	EPSDE Mean error Std	CoDE Mean error Std	MPEDA Mean error Std	SGO Mean error Std
$F_{23}$	5.50e+02 (8.05e+01)–	5.34e+02 (2.77e–04)–	5.34e+02 (3.57e–03)–	8.58e+02 (6.82e+01)–	5.34e+02 (4.12e–04)–	5.34e+02 (3.87e–04)–	5.00e+02 (6.98e+01)
$F_{24}$	2.00e+02 (2.85e–14)≈	2.00e+02 (2.85e–14)≈	2.00e+02 (6.20e–13)≈	2.13e+02 (1.52e+00)–	2.00e+02 (2.85e–14)≈	2.00e+02 (2.21e–14)≈	2.00e+02 (0.00e+00)
$F_{25}$	2.11e+02 (7.92e–01)–	2.11e+02 (7.32e–01)–	2.14e+02 (2.00e+00)–	2.13e+02 (2.55e+00)–	2.11e+02 (9.02e–01)–	2.09e+02 (3.32e–01)–	2.00e+02 (0.00e+00)
–	15	16	16	14	12	13	
+	8	7	7	10	11	10	
≈	2	2	2	1	2	2	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of JADE, jDE, SaDE, EPSDE, CoDE and MPEDA. “–”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

**Table 14** Experimental results of CLPSO, CMA-ES, GL-25, TLBO and SGO over 25 independent runs on 25 test functions of 30 variables with 300,000 FEs

Function	CLPSO Mean error ± Std	CMA-ES Mean error ± Std	GL-25 Mean error ± Std	TLBO Mean error ± Std	SGO Mean error ± Std
$F_1$	0.00e+00 ± 0.00e+00≈	1.58e–25 ± 3.35e–26–	5.60e–27 ± 1.76e–26–	0.00e+00 ± 0.00e+00≈	0.00e+00 ± 0.00e+00
$F_2$	8.40e+02 ± 1.90e+02–	1.12e–24 ± 2.93e–25+	4.04e+01 ± 6.28e+01–	0.00e+00 ± 0.00e+00+	6.79e–09 ± 4.79e–09
$F_3$	1.42e+07 ± 4.19e+06–	5.54e–21 ± 1.69e–21+	2.19e+06 ± 1.08e+06–	1.82e+05 ± 4.58e+03≈	1.65e+05 ± 9.07e+04
$F_4$	6.99e+03 ± 1.73e+03–	9.15e+05 ± 2.16e+06–	9.07e+02 ± 4.25e+02–	0.00e+00 ± 0.00e+00+	2.06e+01 ± 1.26e+01
$F_5$	3.86e+03 ± 4.35e+02–	2.77e–10 ± 5.04e–11+	2.51e+03 ± 1.96e+02–	4.60e+03 ± 1.98e+03–	2.08e+03 ± 4.81e+02
$F_6$	4.16e+00 ± 3.48e+00–	4.78e–01 ± 1.32e+00+	2.15e+01 ± 1.17e+00–	3.47e+01 ± 2.42e+01–	2.63e+00 ± 1.74e+00
$F_7$	4.51e–01 ± 8.47e–02–	1.82e–03 ± 4.33e–03+	2.78e–02 ± 3.62e–02–	1.63e–02 ± 1.89e–02–	7.00e–03 ± 5.60e–03
$F_8$	2.09e+01 ± 4.41e–02–	2.03e+01 ± 5.72e–01≈	2.09e+01 ± 5.94e–02–	2.08e+01 ± 4.90e–02≈	2.08e+01 ± 5.83e–03
$F_9$	0.00e+00 ± 0.00e+00+	4.45e+02 ± 7.12e+01–	2.45e+01 ± 7.35e+00+	2.30e+01 ± 1.14e+00+	9.27e+01 ± 2.15e+01
$F_{10}$	1.04e+02 ± 1.53e+01–	4.63e+01 ± 1.16e+01+	1.42e+02 ± 6.45e+01–	1.09e+02 ± 4.02e+01–	9.60e+01 ± 1.18e+01
$F_{11}$	2.60e+01 ± 1.63e+00–	7.11e+00 ± 2.14e+00+	3.27e+01 ± 7.79e+00–	1.77e+01 ± 2.71e+00–	1.72e+01 ± 1.54e+00
$F_{12}$	1.79e+04 ± 5.24e+03–	1.26e+04 ± 1.74e+04–	6.53e+04 ± 4.69e+04–	1.84e+04 ± 2.17e+04–	4.50e+02 ± 2.12e+01
$F_{13}$	2.06e+00 ± 2.15e–01+	3.43e+00 ± 7.60e–01–	6.23e+00 ± 4.88e+00–	3.01e+00 ± 1.02e+00+	3.26e+00 ± 4.57e–01
$F_{14}$	1.28e+01 ± 2.48e–01–	1.47e+01 ± 3.31e–01–	1.31e+01 ± 1.84e–01–	1.30e+01 ± 4.27e–01–	1.17e+01 ± 3.29e–01
$F_{15}$	5.77e+01 ± 2.76e+01+	5.55e+02 ± 3.32e+02–	3.04e+02 ± 1.99e+01–	2.80e+02 ± 7.48e+01–	2.75e+02 ± 6.56e+01
$F_{16}$	1.74e+02 ± 2.82e+01–	2.98e+02 ± 2.08e+02–	1.32e+02 ± 7.60e+01–	2.31e+02 ± 1.17e+02–	1.12e+02 ± 6.22e+01
$F_{17}$	2.46e+02 ± 4.81e+01–	4.43e+02 ± 3.34e+02–	1.61e+02 ± 6.80e+01–	2.73e+02 ± 1.21e+01–	1.62e+02 ± 5.91e+00
$F_{18}$	9.13e+02 ± 1.42e+00–	9.04e+02 ± 3.01e–01–	9.07e+02 ± 1.48e+00–	9.08e+02 ± 4.90e–01–	9.00e+00 ± 0.00e+00
$F_{19}$	9.14e+02 ± 1.45e+00–	9.16e+02 ± 6.03e+01–	9.06e+02 ± 1.24e+00–	9.09e+02 ± 8.00e–01–	9.00e+00 ± 0.00e+00
$F_{20}$	9.14e+02 ± 3.62e+00–	9.04e+02 ± 2.71e–01–	9.07e+02 ± 1.35e+00–	9.07e+02 ± 2.24e+00–	9.00e+00 ± 0.00e+00
$F_{21}$	5.00e+02 ± 3.39e–13≈	5.00e+02 ± 2.68e–12≈	5.00e+02 ± 4.83e–13≈	5.01e+02 ± 1.96e+00–	4.79e+02 ± 2.03e+01
$F_{22}$	9.72e+02 ± 1.20e+01–	8.26e+02 ± 1.46e+01–	9.28e+02 ± 7.04e+01–	8.89e+02 ± 1.76e+01–	4.68e+02 ± 2.12e+01
$F_{23}$	5.34e+02 ± 2.19e–04–	5.36e+02 ± 5.44e+00–	5.34e+02 ± 4.66e–04–	5.39e+02 ± 8.45e+00–	5.00e+02 ± 6.98e+01
$F_{24}$	2.00e+02 ± 1.49e–12≈	2.12e+02 ± 6.00e+01–	2.00e+02 ± 5.52e–11≈	2.01e+02 ± 4.00e–01–	2.00e+02 ± 0.00e+00
$F_{25}$	2.00e+02 ± 1.96e+00–	2.07e+02 ± 6.07e+00–	2.17e+02 ± 1.36e–01–	2.00e+02 ± 0.00e+00≈	2.00e+02 ± 0.00e+00
–	16	16	22	17	
+	6	7	1	4	
≈	3	2	2	4	

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of CLPSO, CMA-ES and GL-25. “–”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively



have considered population size to 100, and FEs is same with other methods and is 300,000. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance level has been conducted on the experimental results, and the last three rows of Tables 13 and 14 summarize the experimental results.

According to Wilcoxon's rank-sum test, it is clear that SGO performs better than JADE in 15 test functions and equivalent to 2 test functions out of 25 test functions. It is better than jDE in 16 test functions and equivalent to 2 test functions out of 25 test functions. It can be seen that compared to SaDE, it performs better in 16 test functions and equivalent to 2 test functions out of 25 test functions. SGO is better than EPSDE in 14 test functions and equivalent to 1 test function out of 25 functions. From the table, it can be verified that SGO is better than CoDE in 12 test functions and equivalent to 2 test functions out of 25 test functions, again better than MPEDE in 13 test functions and equivalent to 2 test functions out of 25 test functions and better than CLPSO in 16 test functions and equivalent to 3 test functions out of 25 test functions. Compared to CMA-ES, our proposed technique is better in 16 test functions and equivalent to 2 test functions out of 25 test functions. We have verified that SGO is better than GL-25 in 22 test functions and equivalent to 2 test functions out of 25 functions, and better than TLBO in 17 test functions and equivalent to 4 test functions out of 25 test functions. So, it is interesting to note that the performance of SGO is better than other algorithms according to Wilcoxon's rank-sum test.

#### Experiment 10: SGO vs. PSO, CPSO, CLPSO, CMA-ES, G3-PCX, DE, and TLBO using composite functions

In this experiment, we have considered six composite test functions and eight novel algorithms, particle swarm optimizer (PSO) [7], cooperative PSO (CPSO) [54], comprehensive learning PSO (CLPSO) [28], evolution strategy with covariance matrix adaptation (CMA-ES) [55], G3 model with PCX crossover (G3-PCX) [56], differential evolution (DE) [2], teaching–learning-based optimization [24] and proposed SGO for testing their performances. The detailed descriptions of these functions are given in papers [57] and [25], and the algorithms in their respective papers. Parameter settings for the composite functions are as in [25].

Table 15 shows the results obtained using the eight algorithms on six composite functions. For each test function, each algorithm is run 20 times, and the maximum fitness evaluations are 50,000 for all algorithms. For our proposed algorithm, we have considered population size as 100. The mean values of the results are recorded in Table 15. The best optimal values are shown in bold face. To have statistically sound conclusions, Wilcoxon's rank-sum test at a 0.05 significance

level has been conducted on the experimental results, and the last three rows of Table summarize the experimental results.

According to Wilcoxon's rank-sum test, it is clear that SGO performs better than PSO in six test functions out of all six test functions, better than CPSO in six test functions out of all six test functions but better than CLPSO in three test functions out of six test functions. SGO is better than CMA-ES in six test functions out of all six functions, better than G3-PCX in five test functions out of six test functions but better than DE in three test functions and equivalent to one test function out of six test functions; whereas it is better than TLBO in three test functions out of six test functions. So, from Table 15, it is clear that out of seven algorithms, in all cases except CLPSO and TLBO, SGO is showing better result; however, with CLPSO and TLBO, SGO is showing equivalent result. From the last column of Table 15, it is also clear that SGO sometimes reaches to optimal solution.

#### Conclusion

This paper proposes a new efficient optimization algorithm that is inspired by the social behavior of humans toward solving a complex problem. Whenever a problem/task has been solved by a single person, it becomes too difficult to solve or the problem may remain unsolvable. But when the same problem has been solved by a group of persons, the difficulty becomes easy and the unsolvable problem may become solvable. In a social group, people are influenced by the characteristics (i.e., traits) of the successful person, and eventually, they also change/modify their traits accordingly and become capable to solve/address complex problems/situations. This concept has motivated us to propose a new optimization algorithm known as social group optimization (SGO). The concept and the mathematical formulation of SGO algorithm are explained in this paper with a flowchart. To judge the effectiveness of SGO, extensive experiments have been conducted on number of different unconstrained benchmark functions as well as 25 standard numerical benchmark functions taken from the IEEE Congress on Evolutionary Computation 2005 competition. Performance comparisons are made with state-of-the-art optimization techniques like GA, PSO, DE, ABC and its variants and the recently developed TLBO. Different variants of the popular evolutionary optimization techniques are also taken into consideration for comparing them with SGO. The experimental results show that the proposed social group optimization outperforms all investigated optimization techniques in computational costs and also provides optimal solutions for most of the considered functions. One of the best things in this algorithm is that it is easier to understand and to implement in comparison to other algorithms and their variants. It remains to see how SGO works for multi-objective optimization problems in future.

**Table 15** Performance comparison of algorithms PSO, CPSO, CLPSO, CMA-ES, G3-PCX, DE, TLBO and SGO using composite functions

Composite functions	PSO	CPSO	CLPSO	CMA-ES	G3-PCX	DE	TLBO	SGO	Best value in SGO
CF1									
Mean	1.00e+02–	1.56e+02–	5.73e–08–	1.00e+02–	6.00e+01–	6.75e–02–	3.11e–01–	1.54e–25	0
Std	8.17e+02	1.34e+02	1.92e+01	1.89e+02	6.99e+01	1.11e–01	3.04e–01	1.51e–25	
CF2									
Mean	1.56e+02–	2.42e+02–	1.92e+01+	1.62e+02–	9.26e+01–	2.87e+01+	1.70e+01+	4.13e+01	0
Std	1.32e+02	1.49e+02	1.47e+01	1.51e+02	9.91e+01	8.62e+01	7.22e+00	4.38e+01	
CF3									
Mean	1.72e+02–	3.63e+02–	1.33e+02+	2.14e+02–	3.20e+02–	1.44e+02≈	1.24e+02+	1.40e+02	0
Std	3.29e+01	1.96e+02	2.00e+01	7.42e+01	1.25e+02	1.94e+01	6.04e+01	4.07e+01	
CF4									
Mean	3.14e+02–	5.22e+02–	3.22e+02–	6.16e+02–	4.93e+02–	3.25e+02–	2.94e+02–	2.82e+02	2.23e+02
Std	2.00e+01	1.22e+02	2.75e+01	6.72e+02	1.42e+02	1.47e+01	3.15e+01	2.90e+01	
CF5									
Mean	8.35e+01–	2.56e+02–	5.37e+00–	3.59e+02–	2.60e+01–	1.08e+01–	5.18e+00–	2.39e+00	0
Std	1.00e+02	1.76e+02	2.60e+00	1.69e+02	4.16e+01	2.60e+00	1.62e+00	1.07e+00	
CF6									
Mean	8.61e+02–	8.53e+02–	5.01e+02+	9.00e+02–	7.72e+02+	4.91e+02+	2.30e+02+	7.44e+02	4.01e+02
Std	1.26e+02	1.28e+02	7.78e–01	8.32e–02	1.89e+02	3.95e+02	4.84e+01	1.89e+02	
–	06	06	03	06	05	03	03		
+	00	00	03	00	01	02	03		
≈	00	00	00	00	00	01	00		

Wilcoxon's rank-sum test at a 0.05 significance level is performed between SGO and each of PSO, CPSO, CLPSO, CMA-ES, G3-PCX, DE, TLBO. “–”, “+”, and “≈” denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of SGO, respectively

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## Appendix: Benchmark functions

All problems are divided into four categories such as US, MS, UN, MN, and its range, formulation, characteristics and the dimensions of these problems are listed in the following Table 16.

**Table 16** Benchmark functions used in experiments 1

Sl. no.	Function	$D$	$C$	Range	Formulation	Value
1	Step	30	US	$[-100, 100]$	$f(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	$f_{\min} = 0$
2	Sphere	30	US	$[-100, 100]$	$f(x) = \sum_{i=1}^D x_i^2$	$f_{\min} = 0$
3	Sum Squares	30	US	$[-10, 10]$	$f(x) = \sum_{i=1}^D i x_i^2$	$f_{\min} = 0$
4	Quartic	30	US	$[-1.28, 1.28]$	$f(x) = \sum_{i=1}^D i x_i^4 + \text{random}(0, 1)$	$f_{\min} = 0$
5	Beale	2	UN	$[-4.5, 4.5]$	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	$f_{\min} = 0$
6	Easom	2	UN	$[-100, 100]$	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	$f_{\min} = -1$
7	Matyas	2	UN	$[-10, 10]$	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	$f_{\min} = 0$
8	Zakharov	10	UN	$[-5, 10]$	$f(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5i x_i)^2 + (\sum_{i=1}^D 0.5i x_i)^4$	$f_{\min} = 0$
9	Powell	24	UN	$[-4, 5]$	$f(x) = \sum_{i=1}^{D/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	$f_{\min} = 0$
10	Schwefel 1.2	30	UN	$[-100, 100]$	$f(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	$f_{\min} = 0$
11	Schwefel 2.21	30	UN	$[-100, 100]$	$f(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	$f_{\min} = 0$
12	Schwefel 2.22	30	UN	$[-10, 10]$	$f(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	$f_{\min} = 0$
13	Bohachevsky1	2	MS	$[-100, 100]$	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	$f_{\min} = 0$
14	Bohachevsky2	2	MS	$[-100, 100]$	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) * \cos(4\pi x_2) + 0.3$	$f_{\min} = 0$
15	Bohachevsky3	2	MS	$[-100, 100]$	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos((3\pi x_1) + (4\pi x_2)) + 0.3$	$f_{\min} = 0$
16	Booth	2	MS	$[-10, 10]$	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$f_{\min} = 0$
17	Rastrigin	30	MS	$[-5.12, 5.12]$	$f(x) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$	$f_{\min} = 0$
18	Noncontinuous rastrigin	30	MS	$[-5.12, 5.12]$	$f(x) = \sum_{i=1}^D [y_i^2 - 10\cos(2\pi y_i) + 10]$ where $y_i = \begin{cases} x_i &  x_i  < 0.5 \\ \text{round}(2x_i)/2 &  x_i  \geq 0.5 \end{cases}$	$f_{\min} = 0$
19	Six Hump Camel Back	2	MN	$[-5, 5]$	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$f_{\min} = -1.03163$
20	Griewank	30	MN	$[-600, 600]$	$f(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	$f_{\min} = 0$
21	Ackley	30	MN	$[-32, 32]$	$f(x) = -20\exp(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^D \cos(2 * \pi i * x_i)) + 20 + e$	$f_{\min} = 0$
22	Multimod	30		$[-10, 10]$	$f(x) = \sum_{i=1}^D  x_i  \prod_{i=1}^D  x_i $	$f_{\min} = 0$
23	Weierstrass	30		$[-0.5, 0.5]$	$f(x) = \sum_{i=1}^D (\sum_{k=0}^{k_{\max}} [a\cos(2\pi b^k(x_i + 0.5))]) - D \sum_{k=0}^{k_{\max}} [a\cos(2\pi b^k(x_i + 0.5))]$ , where $a = 0.5, b = 3, k_{\max} = 20$	$f_{\min} = 0$
24	Elliptic	30		$[-100, 100]$	$\sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$	$f_{\min} = 0$
25	Rosenbrocks	30	UN	$[-30, 30]$	$\sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$f_{\min} = 0$

$D$  dimension,  $C$  characteristic,  $U$  unimodal,  $M$  multimodal,  $S$  separable,  $N$  non-separable

## References

1. Tang KS et al (1996) Genetic algorithms and their applications. *IEEE Signal Process Mag* 13:22–37
2. Storn R, Price K (2010) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 23:689–694
3. Passino K (2002) Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst Mag* 22(3):52–67
4. Eusuff M et al (2006) Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Eng Optim* 38(2):129–154
5. Johnson DS et al (1989) Optimization by simulated annealing. An experimental evaluation Part I. Graph partitioning. *Oper Res* 37:865–892
6. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 12:53–66
7. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proc. IEEE Congr. Evol. Comput. Australia*, pp 1942–1948
8. Karaboga D (2005) An idea based on honeybee swarm for numerical optimization. Technical Report TR06. Erciyes University, Engineering Faculty, Computer Engineering Department
9. Rashedi E et al (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248
10. Yang XS, Deb S (2010) Engineering optimisation by cuckoo search. *Int J Math Modell Numer Optim* 1(4):330–343
11. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12:702–713
12. Hosseini HS (2007) Problem solving by intelligent water drops. In: *IEEE Congress on Evolutionary Computation, CEC vol 2007*, pp 3226–3231
13. Bramer M et al. (2010) Firefly algorithm, Lévy flights and global optimization, *Research and Development in Intelligent Systems*, vol XXVI, Springer, London, pp 209–218
14. Fathian M et al (2007) Application of honey-bee mating optimization algorithm on clustering. *Appl Math Comput* 190:1502–1513
15. Marinakis Y et al (2011) Honey bees mating optimization algorithm for the Euclidean traveling salesman problem. *Inf Sci* 181:4684–4698
16. Gonzalez J, Pelta D, Cruz C, Terrazas G, Krasnogor N, Yang X-S (2011) A new metaheuristic bat-inspired algorithm. In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, Berlin/Heidelberg, pp 65–74
17. Geem Z, Yang X-S (2009) Harmony search as a metaheuristic algorithm. In: *Music-Inspired Harmony Search Algorithm*. Springer, Berlin/Heidelberg, pp 1–14
18. Erol OK, Eksin I (2006) A new optimization method: big bang-big crunch. *Adv Eng Softw* 37:106–111
19. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. *Inf Sci* 222:175–184
20. Rao RV et al (2011) teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Elsevier Comput Aided Des* 43:303–315
21. Suganthan PN et al (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, Nanyang Technol. Univ., Singapore, IIT Kanpur, Kanpur, India, Tech. Rep. KanGAL #2005005, May 2005
22. Karaboga Dervis, Akay Bahriye (2009) A comparative study of Artificial Bee Colony algorithm. *Appl Math Comput* 214:108–132
23. Karaboga D, Akay B (2009) Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization. In: *Proceedings of IPROMS-2009 on Innovative Production Machines and Systems*, Cardiff, UK
24. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching learning based optimization: an optimization method for continuous non-linear large scale problems. *Inf Sci* 183:1–15
25. Rao RV, Waghmare GG (2013) Solving Composite Test Functions Using Teaching-Learning-Based Optimization Algorithm. In: *Proc. of Int. of front. In Intelligent Computing*, Springer AISC 199, pp 395–403,
26. Ratnaweera A, Halgamuge S, Watson H (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8:240–255
27. Zhan ZH, Zhang J, Li Y, Chung SH (2009) Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern B Cybern* 39:1362–1381
28. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
29. Zhan Z-H, Zhang J, Li Y, Shi Y-H (2011) Orthogonal learning particle swarm optimization. *IEEE Trans Evol Comput* 15(6):832–847
30. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417
31. Brest J et al (2006) Selfadapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10(6):646–657
32. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. *IEEE Trans Evol Comput* 13(5):945–958
33. Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans Evol Comput* 15(1):55–66
34. Mallipeddi R et al (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11(2):1679–1696
35. Alatas B (2010) Chaotic bee colony algorithms for global numerical optimization. *Expert Syst Appl* 37:5682–5687
36. Zhu GP, Kwong S (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl Math Comput* 217:3166–3173
37. Kang F, Li JJ, Ma ZY (2011) Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Inform Sci* 12:3508–3531
38. Gao Weifeng, Liu Sanyang (2011) Improved artificial bee colony algorithm for global optimization. *Inf Process Lett* 111:871–882
39. Gao W, Liu S, Huang L (2012) A global best artificial bee colony algorithm for global optimization. *J Comput Appl Math* 236:2741–2753
40. Chen S-M, Sarosh A, Dong Y-F (2012) Simulated annealing based artificial bee colony algorithm for global numerical optimization. *Appl Math Comput* 219:3575–3589
41. Imanian N, Shiri ME, Moradi P (2014) Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems. *Eng Appl Artif Intell* 36:148–163
42. Chuang L-Y, Hsiao C-J, Yang C-H (2011) Chaotic particle swarm optimization for data clustering. *Expert Syst Appl* 38:14555–14563
43. Yang C-H, Tsai S-W, Chuang L-Y, Yang C-H (2012) An improved particle swarm optimization with double-bottom chaotic maps for numerical optimization. *Appl Math Comput* 219:260–279
44. Sun S, Li J (2014) A two-swarm cooperative particle swarms optimization. *Swarm Evol Comput* 15:1–18
45. Shi, YH, Eberhart RC (1998) A modified particle swarm optimizer. In: *Proc. IEEE World Congr. Comput. Intell.* May 1998, pp 69–73
46. Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: *Proc. IEEE Congr. Evol. Comput.*, July 1999, pp 1931–1938

47. Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. *IEEE Trans Evol Comput* 8(3):204–210
48. Bratton D, Kennedy J (2007) Defining a standard for particle swarm optimization. In: *Proc. IEEE Swarm Intel. Symp.* April 2007, pp 120–127
49. Ren Z, Zhang A, Wen C, Feng Z (2014) A scatter learning particle swarm optimization algorithm for multimodal problems. *IEEE Trans Cybern* 44(7):1127–1140
50. Wu G, Qiu D, Yu Y, Pedrycz W, Ma M, Li H (2014) Superior solution guided particle swarm optimization combined with local search techniques. *Expert Syst Appl* 41:7536–7548
51. Wu G, Mallipeddi R, Suganthan PN, Wang R, Chen H (2015) Differential evolution with multi-population based ensemble of mutation strategies. *Inf Sci.* doi:[10.1016/j.ins.2015.09.009](https://doi.org/10.1016/j.ins.2015.09.009)
52. Hansen N, Ostermeier A (2001) Completely derandomized self adaptation in evolution strategies. *Evol Comput* 9(2):159–195
53. Garcia-Martinez C, Lozano M, Herrera F, Molina D, Sanchez AM (2008) Global and local real-coded genetic algorithms based on parent-centric crossover operators. *Eur J Oper Res* 185(3):1088–1113
54. van den Berghm F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput* 8(3):225–239
55. Hansen N, Muller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol Comput* 11(1):1–18
56. Deb K, Anand A, Joshi D (2002) A computationally efficient evolutionary algorithm for real-parameter optimization. *KanGAL Report No. 2002003*, April 2002
57. Liang JJ, Suganthan PN, Deb K (2005) Novel composition test functions for numerical global optimization. *IEEE Trans Evol Comput* 5(1):1141–1153