Applied Soft Computing Journal 95 (2020) 106524

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Modified Social Group Optimization—a meta-heuristic algorithm to solve short-term hydrothermal scheduling

Anima Naik [a,*], Suresh Chandra Satapathy [b], Ajith Abraham [c]

[a] Department of CSE, KL University, Hyderabad, Telangana, India
[b] School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India
[c] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, WA 98071, USA

ARTICLE INFO

Article history:
Received 26 February 2020
Received in revised form 19 June 2020
Accepted 3 July 2020
Available online 16 July 2020

Keywords:
Meta-heuristic algorithm
Benchmark functions
Fitness function evaluations
Social Group Optimization
Hydrothermal scheduling problem

ABSTRACT

Social Group Optimization (SGO), developed by Satapathy et al. in the year 2016, is a class of meta-heuristic optimization inspired by social behavior. It has two phases: improving phase and acquiring phase. In the improving phase, each individual improves its knowledge by interacting with the best person/solution and in acquiring phase, the individuals interact with randomly selected individuals and the best person simultaneously to acquire knowledge. Modified Social Group Optimization (MSGO) is the improved version of SGO, where the acquiring phase is modified. A self-awareness probability factor is added in the acquiring phase, which enhances the learning capability of an individual from the best-learned person in the societal setup. It is observed that this modification has improved both exploration and exploitation abilities in comparison with the conventional SGO. To analyze the performance of the MSGO, an exhaustive performance comparison is made with GA, PSO, DE, ABC, and a few newer algorithms of the years 2010–2019. The results are tabulated in six experiments. Later, MSGO is applied to solve the short-term hydrothermal scheduling (HTS) problem. The central objective of the HTS problem is to ascertain the optimal plan of action for hydro and thermal generation minimizing the fuel cost of thermal plants and, at the same time satisfying various operational and physical constraints. The valve point loading effect related to the thermal power plants, transmission loss, and other constraints lead HTS as a complex non-linear, non-convex, and non-smooth optimization problem. Simulation results clearly show that the MSGO method is capable of obtaining a better solution.

1. Introduction

The meta-heuristic optimization algorithms which are based on simulating nature and artificial intelligence toll are widely used for solving global optimization problems. Two important characteristics of such algorithms named "exploration" and "exploitation" help diversifying the search space and intensify search in some promising areas [1]. From the 1960s to date, numbers of meta-heuristic optimization algorithms have been proposed. *According to the" No-Free-Lunch (NFL) [2] theorem for optimization, none of the algorithms can solve all optimization problems. This theorem makes this area of research open, in which researchers are allowed to improve/adapt/hybridize current algorithms for solving different problems or propose new algorithms for providing competitive results compared to current algorithms"*. This fact resulted in the development of numerous meta-heuristic optimization algorithms over the years. Researchers have developed new algorithms or have modified existing ones by trying to improve the original body of the algorithms or by hybridizing with other algorithms. However, there is no concrete proof so far that anyone particular algorithm is suitable/applicable to all class of problems. Social Group Optimization (SGO) [3] is a class of meta-heuristic optimization algorithm inspired by the social behavior of an individual in a group to solve complex problems. The knowledge of each individual is mapped by its fitness. The algorithm involves two phases: improving phase and acquiring phase. In the improving phase, each individual improves knowledge by interacting with the best person (best solution) and in acquiring phase, the individuals interact with randomly selected individuals and the best person simultaneously to acquire knowledge. In this work, the performance of the existing SGO algorithm is improved by incorporating a probabilistic factor known as self-awareness probability in the acquiring phase, thus proposing a modified algorithm of SGO. It is coined as the MSGO algorithm.

Any new modification to an algorithm mandates exhaustive comparisons with existing algorithms. This is more appropriate

* Corresponding author.
E-mail addresses: animanaik@klh.edu.in (A. Naik), suresh.satapathyfcs@kiit.ac.in (S.C. Satapathy), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.asoc.2020.106524
1568-4946/

to meta-heuristic algorithms since in recent times, several new algorithms are in place, and even several revisions of well known classical meta-heuristic algorithms are also existing. In this work, before applying MSGO to a real-world problem i.e., HTS problem [4], we have undertaken a very intense comparison with GA, PSO, DE, ABC, and many recent algorithms from 2010–2019 on several benchmark functions and tabulated the results and ranked the algorithms.

The remaining paper is organizing as follows: In Section 2, a comprehensive review of meta-heuristic algorithms is presented. SGO algorithm is briefly summarized in Section 3. Section 4 discusses the modified SGO. In Section 5, comparisons are done with other meta-heuristic algorithms and experimented with HTS and all results are tabulated. Section 6 conclusion of the work is presented.

2. Literature review

Optimization methods play a leading role in solving engineering problems. Deterministic methods [5] are computationally costly and may not yield an efficient solution in solving complex nonlinear and multimodal problems. But real-world problems are complex nonlinear and multimodal. Meta-heuristic algorithms are efficient algorithms for solving these problems as they are stochastic in nature and derivative-free. These Meta-heuristic algorithms are inspired by nature to solve complex problems [6,7].

Meta-heuristic algorithms can be classified mainly into four categories: (1) evolutionary-based algorithm [8] (2) swarm intelligence based algorithm [9,10] (3) human-based algorithm [11] and (4) physical and chemical-based algorithm [12]. Evolutionary algorithms mimic concepts of evolution in nature. Swarm intelligence algorithms mimic the intelligence of swarms. Each swarm consists of a group of creatures. So these algorithms originate from the collective behavior of a group of creatures in the swarm. Human-based algorithms are inspired by the behavior of humans. Physical and chemical-based algorithms are inspired by physical rules and chemical reactions of the universe. A flowchart of broad classification is given in Fig. 1. Later, some discussions on a few meta-heuristic algorithms which are widely used and popular are presented in the paper without loss of generality. Authors have picked up a few algorithms only to demonstrate the developments done on each to demonstrate the type of research carried on the meta-heuristic techniques. Further literature of each algorithm mentioned in this section can be referred from the cited reference.

The Genetic Algorithm (GA) [12] is the best example of an evolutionary algorithm that simulates the concepts of Darwinian Theory of Evolution. Originally PSO [21] is proposed by Kennedy and Eberhart in 1995. This algorithm mimics the behavior of flocking birds or a school of fish. In PSO, a swarm of birds/fish representing the candidate solutions travel through the sample space driven by their own and best performances of their neighbors. Although popular, PSO suffers from getting stuck in local minima and premature convergence for certain complex problems. So to overcome this limitation, PSO variants have been developed. Few well known PSO variants are PSO with inertia weight (PSO-w) [93], PSO with constriction factor (PSO-cf) [94], Local version of PSO with inertia weight (PSO-w-local), Local version of PSO with constriction factor (PSO-cf-local) [95], Unified PSO (UPSO) [96], Fully Informed Particle Swarm (FIPS) [97], Fitness-Distance Ratio-based PSO (FDR-PSO) [98], Comprehensive Learning PSO CLPSO [99].

Differential evolution (DE) [19] algorithm is a powerful, straightforward population-based optimization technique mimicking the basic rules of genetics, i.e., mutation and crossover strategies. Depending upon the variants of mutation and crossover strategies, variants of modified DE algorithms are also proposed. The Adaptive DE algorithm (JDE) [100] is a modified DE algorithm with self tuned control parameters and the Self-Adaptive Differential Evolution algorithm (SADE) [101], both, the control parameters as well as the learning strategies of DE are self-adapted during the evolution phase. Similarly, JADE [102], Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies (EPSDE) [103], Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters CoDE [104], Adaptive CoDE [104] are also modified versions of the DE.

The artificial bee colony algorithm (ABC) [23] is an optimizer based on the collaborative and intelligent behavior of a swarm of honey bees. It mimics the foraging behavior of a swarm of bees, which comprises employed bees, onlookers, and scout bees. It is a powerful meta-heuristic algorithm that effectively employs four different selection processes. A global selection process, local selection process carried out by employed and onlooker bees, a local greedy selection process to memorize the best candidate solution, and a random selection process carried out by the scouts.

Teaching Learning Based Optimization (TLBO) [43] an optimization method mimics the influence of a teacher on the learning outcome of its students. The methodology works in two parts: the teacher phase and the learner phase. During the teacher phase, the learner learns or increases their knowledge from the teacher, and in the learner phase, learners increase their knowledge through interaction among themselves. There are variants of TLBO that have been proposed to improve the performance of TLBO. These are Improved TLBO (ITLBO) [105], Orthogonal TLBO (OTLBO) [106], Weighted TLBO (WTLBO) [107], Modified TLBO(MTLBO) [108,109], Cooperative TLBO (CoTLBO) [110]. The applications of these modifications are shown in the area of data clustering [111].

Ant Lion Optimizer (ALO) [33] algorithm mimics the hunting mechanism of ant lions in nature. Five main steps of hunting prey such as the random walk of ants, building traps, entrapment of ants in traps, catching prey, and re-building traps are utilized. Dragonfly Algorithm (DA) [34] is inspired by the static and dynamic swarming behaviors of dragonflies in nature. Here the exploration and exploitation search of optimization is designed by modeling the social interaction of dragonflies in navigating, searching for foods, and avoiding enemies when swarming dynamically or statistically. Gray Wolf Optimization (GWO) [35] is inspired by the hunting behavior of the gray wolves. The individuals update their position concerning the top three best solutions named alpha, beta and delta. First, the next position is calculated about each better solution, and then the individual jumps to the mean of independently calculated position. The hunting behavior of humpback whales inspires the Whale-Optimization algorithm (WOA) [36]. WOA mimics three behaviors of whales known as searching the prey, encircling the prey, and the bubble-net attacking method. Encircling behavior is modeled by searching around the best solution in a circular way. The attacking behavior is modeled by searching the space around the best solution in a spiral form. Finally, the searching behavior is modeled by exploring the region around a randomly selected solution. Self-adaptive parameters are designed to decide the behavior of an individual.

The Grasshopper-Optimization algorithm (GOA) [39] algorithm mimics the behavior of grasshopper swarms and their social interaction. The Crow Search Algorithm (CSA) [37] mimics the behavior of crow birds and their social interaction. The Salp-Swarm Algorithm(SSA) [38] is inspired by the swarming behavior of salps when navigating and foraging in oceans. The Butterfly Optimization algorithm(BOA) [40]mimics the food search and mating behavior of butterflies. The Squirrel Search Optimization

Fig. 1. Classification of meta-heuristic algorithms (see [3,13–92]).

Algorithm (SSOA) [41] imitates the dynamic foraging behavior of southern flying squirrels and their efficient way of locomotion. Harris Hawks Optimizer (HHO) [42] is inspired from the cooperative behaviors of one of the most intelligent birds, Harris' Hawks, in hunting escaping preys (rabbits in most cases) The main inspiration of HHO is the cooperative behavior and chasing style of Harris' hawks in nature called surprise pounce. In this intelligent strategy, several hawks cooperatively pounce prey from different directions in an attempt to surprise it. Harris hawks can reveal a variety of chasing patterns based on the dynamic nature of scenarios and escaping patterns of the prey.

The Volleyball Premier League Algorithm (VPL) [70] is inspired by competition and interaction among volleyball teams during a season. It also mimics the coaching process during a volleyball match. The Socio Evolution & learning optimization algorithm (SELO) [71] is inspired by the social learning behavior of humans organized as families in a societal setup. Gravitational Search Algorithm (GSA) [75] is based on the law of gravity and mass. Here search agents are a collection of masses that interact with each other based on the Newtonian gravity and the laws of motion. The Sine Cosine Algorithm (SCA) [88] algorithm is based on the concepts that it creates multiple initial random candidate solutions and requires them to fluctuate outwards or towards the best solution using a mathematical model based on sine and cosine functions. The Multi-Verse Optimizer (MVO) [89] is based on three concepts in cosmology: white hole, black hole, and wormhole Lightning Search Algorithm (LSA) [87] are inspired by lightning phenomena The Lightning Attachment Procedure Optimization (LAPO) [90] algorithm is based on the concepts of the lightning attachment procedure. The inspiration of both LSA and LAPO is the same; however, the view and the equations by which the solutions are updated, are entirely different. In LSA, the solutions are the atoms traveling through the atmosphere by their kinetic energy and ionizing the nearby space by collision with other molecules and atoms. In this method, only the downward movement of the ionized channel is considered. In LAPO, the solutions are the jumping points of the lightning. In this model, both upward and downward leaders are considered. Atom Search Optimization (ASO) [91] is inspired by basic molecular dynamics. It mimics the atomic motion model in nature, where atoms interact through interaction forces resulting from the Lennard-Jones potential and constraint forces resulting from the bond-length potential. The Golden Ratio Optimization Method (GROM) [92] is inspired by the golden ratio of plant and animal growth.

Social Group Optimization (SGO) [3] is inspired by the social behavior of an individual in a group to solve complex problems. The knowledge of each individual is mapped by its fitness. The algorithm involves two phases: improving phase and acquiring phase. In the improving phase, each individual improves knowledge by interacting with the best person (best solution). In the acquiring phase, the individuals interact with randomly selected individuals and the best person simultaneously to acquire knowledge. After that, many researchers have explored these optimization techniques for solving different complex problems in different research areas. In the medical field, social group optimization supported as an automated tool to examine skin melanoma in dermoscopy images [112], SGO with Fuzzy-Tsallis entropy helps in the segmentation of ischemic stroke lesion in brain MRI [113]. In the field of electrical science, a transformer fault diagnosis model is introduced using an optimal hybrid dissolved gas analysis features subset with improved social group optimization and support vector machine classifier [114]. SGO algorithm is used to develop a procedure for maximizing the natural frequencies of laminated composite stiffened panels. It optimizes the quantities and sizes of the stiffener to maximize the fundamental frequency of the panels under certain constraints [115]. In a cloud environment, adequate resource allocation and scheduling of tasks are proposed using SGO for minimizing the make-span time and maximizing throughput [116]. The circular array synthesis is performed using social group optimization. The synthesis technique employs both non-uniform amplitudes and non-uniform spacing between the elements [117] SGO is used to solve the optimized problem of economic load dispatch in the operational planning of the power system [118]. SGO has been used in gray and RGB image multi-level pre-processing [119]. The application of SGO can also be shown in the wireless sensor network [120]. Like this, there are so many areas where SGO performs its application in solving problems.

From the studies of various algorithms discussed and cited in Fig. 1, a clear understanding is derived that every algorithm is designed to expand its capabilities of search space to obtain an optimal solution. Each algorithm may have some user-defined and algorithmic parameters to tune. Variants of algorithms are aimed to achieve better performance in terms of qualitative results and computational time. Also, it is observed that all algorithms are not equally good to solve all types of real-world problems. This motivates us to modify further, improve existing algorithms, or develop new algorithms to solve a variety of problems.

There are two important factors related to heuristic optimization methods. The first one is exploration, which refers to searching the whole space and having various solutions in each iteration. This factor shows the capability of a method in a global search. The other factor is exploitation, which refers to the quality of solutions in each iteration. This factor shows the capability of a method in the local search and finding the best answer around a solution. It should be mentioned that these two factors are in contrast to each other. In other words, focusing too much on local search, i.e., exploitation may result in getting stuck in local optimum points, and too much focusing on global search, i.e., exploration, may cause the low quality of the final best answer [121]. So an algorithm should be in the form that it can balance in between and find out an optimal solution to the problem. Hence we have kept a look at the balance between exploration and exploration search capability of the SGO algorithm. It can be observed through experiment 1 of this paper that although exploration and exploration search capability of the SGO algorithm is much better in comparison to many algorithms, it is still unable to find the optimal solution for some fixed-dimensional multimodal functions [36]. For an example of Shekel family (Shekel 5, Shekel 7, Shekel 10). Hence, it is a need to balance between exploration search and exploration search. Motivated from this limitation of the SGO algorithm, we have proposed the modified SGO (SGO), and the performance of MSGO is studied simulating benchmark functions are used by many researchers [35,36,42,70,87–92] and comparing with a few meta-heuristic algorithms listed in Fig. 1. Finally, MSGO is applied to solve the Short-term Hydrothermal Scheduling Problem.

The Short-term Hydrothermal Scheduling (HTS) problem is a complex nonlinear dynamic constrained optimization problem, which plays an important role in the economical operation of electric power systems. Several researchers have extensively investigated the HTS problems for the last few decades. A bibliographical survey on HTS reveals that various numerical classical optimization techniques have been employed to resolve the HTS problem. Because of the highly nonlinear characteristics of the HTS problem with many local optimum solutions and a large number of constraints, these classical methods may not perform satisfactorily in solving HTS problems. The studies on population-based techniques have shown that these methods can be efficiently used to eliminate most of the difficulties of classical methods. In recent years, heuristic methods such as genetic algorithm (GA) [122], simulated annealing (SA) [123], tabu search (TS) [124], ant colony optimization (ACO) [125], particle swarm optimization (PSO) [126–128], differential evolution (DE) [129], quantum-inspired evolutionary algorithm (QEA) [130] and artificial bee colony (ABC) [131] have been employed for solving Short-Term HTS problem. Moreover, some combinatorial methods such as fuzzy satisfying method based on evolutionary programming technique [132] and hybrid differential evolution and sequential quadratic programming (DE–SQP) algorithms [133] have been successfully applied to this problem. These methods can provide a good solution and deal with complicated nonlinear constraints more simply and effectively. However, the methods mentioned above require a large amount of computation time, especially for large-scale HTS problems. Besides, they are inclined to trap into the local optimum in the later evolution period and sensitive to initial points. This problem has been included in the "CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems" [4]. Many algorithms have participated in this competition [134–141]. In this paper, the HTS problem is solved by the modified SGO algorithm and performance compared with other algorithms.

3. Social group optimization (SGO) algorithm

SGO algorithm is based on the social behavior of a human to solve complex problems. Each person represents a candidate solution empowered with some information that has an ability level to solve a problem. The human traits represent the dimension of a person that represents the number of design variables of the problem. This optimization algorithm goes through two-phase: Improving Phase and Acquiring Phase. In improving phase each individual's knowledge (solution) level is improved based on the best individual influence and in the acquiring phase, each individual's knowledge (solution) level is improved by mutual interaction between individuals and also with the best person which one having highest knowledge level and the ability to solve the problem under concern. For a detailed description of the SGO algorithm, please refer to the paper [3,65]. The algorithm of SGO is given below:

Let P_i, $i = 1, 2, 3, \ldots, N$ be the persons of the social group, i.e., the social group contains N persons and each person P_i is defined by $P_i = (p_{i1}, p_{i2}, p_{i3}, \ldots\ldots, p_{iD})$ where D is the number of traits assigned to a person which determine dimensions of a person and f_i, $i = 1, 2, \ldots\ldots, N$ are their corresponding fitness value, respectively.

Improving Phase

The best person in the group (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.

$$[\text{minvalue}, \text{index}] = \min\{f(P_i), i = 1, 2, 3, \ldots\ldots\ldots\ldots, N\}$$
$$\text{gbest} = P(\text{index}, :) \tag{1}$$

for solving the minimization problem

In improving phase, each person gets knowledge from the group's best (gbest) person. The updating of each person can be computed as follows:

Algorithm: The Improving phase

```
For i = 1 : N
    For j=1:D
        Pnew_ij = c * P_ij + rand * (gbest(j) − P_ij)
    End for
End for
Accept Pnew if it gives a better fitness than P
```

where $rand$ is a random number, $rand \sim U(0, 1)$, and c is known as self-introspection parameter lies in between 0 and 1.

Acquiring Phase

In the acquiring phase, a person in a social group interacts with the best person of the group (gbest) and also interacts randomly with other persons of the group for acquiring knowledge. A person acquires new knowledge if other person has more knowledge. The best knowledgeable person 'gbest' has the greatest influence on others to learn from. A person also will 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

$$[\text{minvalue}, \text{index}] = \min\{f(P_i), i = 1, 2, 3, \ldots\ldots\ldots\ldots, N\}$$
$$\text{gbest} = P(\text{index}, :) \tag{2}$$

(P_i's are updated value at the end of improving phase)

Algorithm: The Acquiring phase

```
For i = 1 : N
    Randomly select one person P_r, where i ≠ r
    If f(P_i) < f(P_r)
        For j=1:D
            Pnew_{i,j} = P_{i,j} + rand_1 * (P_{i,j} − P_{r,j}) + rand_2 * (gbest(j) − P_{i,j})
        End for
    Else
        For j=1:D
            Pnew_{i,:} = P_{i,:} + rand_1 * (P_{r,:} − P_{i,:}) + rand_2 * (gbest(j) − P_{ij})
        End for
    End If
End for
Accept Pnew if it gives a better fitness than P.
```

where $rand_1$ and $rand_2$ are two independent random sequences, $rand_1 \sim U(0, 1)$, and $rand_2 \sim U(0, 1)$. These sequences are used to affect the stochastic nature of the algorithm.

4. Modified social group optimization (MSGO) algorithm

Studies on SGO has revealed that though it is performing better compared to many of the meta-heuristic algorithms, still some improvement is warranted to expand its search capability and diversity. Two factors are mainly responsible for achieving this: exploration and exploitation. The authors of the paper have attempted to modify one of the phases of the SGO algorithm to accomplish this. In modified SGO (MSGO), improving phase remains the same, and the acquiring phase only modified. The detail of the modification part is discussed below.

Acquiring Phase

As we know in acquiring phase a person of social group interacts with the best person ($best_P$) 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. The $best_P$ is always best than others, so a person always acquires knowledge from $best_P$. A person acquires something new from other persons if other person has more knowledge, and he or she has a higher self-awareness probability (SAP) to achieve that knowledge. Self-Awareness probability (SAP) defines the ability to acquire a quantity of knowledge from other person. So the modified acquiring phase is expressed as

$$[\text{value}, \text{index\_num}] = \min\{f(P_i), i = 1, 2, 3, \ldots\ldots\ldots\ldots, N\}$$
$$best_P = P(\text{index\_num}, :) \tag{3}$$

for solving minimization problem, where P_i's are updated value at the end of the improving phase.

Algorithm: The Acquiring phase

For i = 1 : N
  Randomly select one person P_r, where $i \neq r$
    If f (P_i) < f (P_r)
      If rand>SAP
        For j=1:D
          $Pnew_{i,j} = P_{i,j} + rand_1 * (P_{i,j} - P_{r,j}) + rand_2 * (best_P(j) - P_{i,j})$
        End for
      Else
        For j=1:D
          $Pnew_{i,:} = lb + rand * (ub - lb)$
        End for
      end if
    Else
      For j=1:D
        $Pnew_{i,j} = P_{i,j} + rand_1 * (P_{r,j} - P_{i,j}) + rand_2 * (best_P(j) - P_{i,j})$
      End for
    End If
End for
Accept $Pnew$ if it gives a better fitness than P

where $rand_1$ and $rand_2$ are two independent random sequences, $rand_1 \sim U(0, 1)$ and $rand_2 \sim U(0, 1)$. These sequences are used to affect the stochastic nature of the algorithm; lb and ub are the lower bound and upper bound of the corresponding design variable.

From intuition, it can be ascertained that higher the value of SAP better the knowledge gain. Hence, it must be higher than 0.5 and less than 1 (i.e., 0.6$\leq$ SAP $\leq$ 0.9). But through the number of experiments, we had concluded that the algorithm shows the best performance when SAP = 0.7. A comprehensive study may be required to find the best parameter setting, which is not the primary concern of this paper. This new change in the acquiring phase of standard SGO has resulted in MSGO. In the next section, the performance of MSGO is demonstrated with several experiments on benchmark functions and comparisons with other meta-heuristic algorithms.

5. Simulation and experimental results

The performance of the Modified Social Group Optimization (MSGO) algorithm is demonstrated in this paper through the number of experiments. A total of seven experiments is carried out in our work to showcase how well the MSGO performs on various types of benchmark functions and a real-world problem. In the first experiment 20 meta-heuristic algorithms such as GA [12], PSO [21], DE [19], ABC [23], WOA [36], SSA [38], MVO [89], SCA [88], LAPO [90], HHO [42], GWO [35], GOA [39], CSA [37], ALO [33], ASO [91], TLBO [43], SGO [3], GSA [75], LSA [87] and DA [34] are considered for comparison purpose from Fig. 1. Twenty-three classical benchmark functions, which are consisting of unimodal, multimodal, and fixed dimensional multimodal, are chosen in common in the first experiment 1. For next five experiments, we have considered benchmark functions that are simulated in respective papers of BOA [40], SSOA [41], VPL [70], SELO [71] and GROM [92]. These five meta-heuristic algorithms are not included in experiment 1, as we have not simulated those algorithms, but the results are reported form their respective papers. And finally, in the last experiment, MSGO is implemented to solve the short-term hydrothermal scheduling problem.

"*For comparing the speed of algorithms, the number of iterations or generations cannot be accepted as a time measure, since algorithms perform the different amount of works in their inner loops, and also they have different population sizes. Hence, we choose max_FEs (maximum Function Evaluations) as a measure of computations time instead of generations or iterations*" [3]. Due to the stochastic nature of meta-heuristic algorithms, we have taken different independent runs (with different seeds of random number generator) for each algorithm and compute best (B), mean(M), and standard deviation(SD) of function values and then tabulated the results. For comparing the performance of algorithms, Wilcoxon's rank-sum (WRS) test [142] at a 0.05 significance level is conducted on experimental results.

Entire experiments are conducted in MATLAB 2016a on an Intel Core i5, 8 GB memory in Windows 10 Environment. The code for the MSGO algorithm will be released at math-works after acceptance of the paper. This specification is needed to be given here so that reader can reproduce the results. Results of experiment 1 are derived after running the codes available from addresses given in APPENDIX B, and for GA, PSO, DE, ABC, TLBO, and SGO, codes are implemented by us. However, we have directly referred the results of the other five algorithms from their respective papers, such as BOA [40], SSOA [41], VP [70], SELO [71] and GROM [92]. Codes for SGO and MSGO are developed by us for simulation purposes. One crucial point can be mentioned here that the performance of any meta-heuristic algorithm is heavily dependent on its parameter settings. There are two types of parameters: Common parameter and specific parameter. Common parameters are Population size (Pop_size) and numbers of Function Evaluations (FEs). Specific parameters are specific to each algorithm. The more the numbers of specific parameters, the more complex it is to tune those. Due to this reason recently, researchers are focusing to suggest specific parameters free meta-heuristic algorithms or trying to suggest adaptable parameters for the algorithm. All these parameters are set according to the details available in respective papers, and for MSGO following settings for specific parameters are chosen.

MSGO setting: For MSGO, there are only two parameters C and SAP called as a self-introspection factor and Self Awareness Probability, respectively. The value of C is set to 0.2 and SAP to 0.7

Experiment 1:

Details:

Total nos of algorithms compared with MSGO: 20

Total nos of benchmark functions: 23 (7 are unimodal benchmark functions, 6 are multimodal benchmark functions, and ten are fixed-dimensional multimodal) given in Appendix A.

The unimodal functions (F1–F7) are suitable for benchmarking the exploitation of algorithms since they have only one global optimum. The multimodal functions (F8–F23), each having a massive number of local optima. They are accommodated to examine the exploration capability of algorithms. All these functions are illustrated in Fig. 2.

This experiment is divided into two tests. Test 1 is on benchmark functions as given in Appendix A whereas in Test 2 high dimension versions of seven unimodal and six multimodal benchmark functions from Appendix A are considered. As the number of function evaluations (FEs) in each iteration for the different algorithm is different (refer Table 1) hence, the number of particles (pop_size) and the number of iterations for each algorithm must have to be selected in such a way that max_FEs would be the same for all algorithms for arriving at a fair comparison.

In Test 1, we have taken pop_size is 30 and max_FEs are 15,000 for all algorithms (here number of iterations are different, for example, for ABC, TLBO, SGO, LAPO, LSA, ALO, MSGO is 250, whereas for others 500). In Test 2, we have taken the number of iterations is 500 and max_FEs are 25,000 for all algorithms. The parameter descriptions of different algorithms are given in Table 1. We have discussed the results of each test in separate headings.

Test 1

In Test 1, we have compared all the 20 algorithms with MSGO algorithms by considering 23 benchmark functions of Appendix A. Each benchmark function is simulated with each algorithm for 32

Table 1
The setting of parameters for algorithms in experiment 1.

Algorithm	Function evaluation in each iteration	Other parameters
GA	1	Real coded, Roulette wheel selection, single point crossover (probability = 1), uniform mutation (probability = 0.01)
PSO	1	Fully connected, cognitive parameter = 2, social parameter = 2, the inertia of weight linearly decreases from 0.9 to 0.4
DE	1	F = 0.5, CR = 0.5
ABC	2	Alimit = 15.
WOA	1	There is no such parameter to set value.
SSA	1	$c_1 = 2^{*}e^{-\left(\frac{4}{L}\right)^2}$, where L = max_iteration = 500.
MVO	1	WEP is increased linearly from 0.2 to 1, and TDR is decreased from 0.6 to 0
SCA	1	There are four parameters r_1, r_2, r_3 and r_4. r_1= a-t*((a)/Max_iter), where a=2, Max_iter = 500, r_2 = (2*pi)*rand, r_3=2*rand, r_4=rand,where rand is random number in the interval (0,1).
LAPO	2	There is no such parameter to set value.
HHO	1	As per paper [42] for HHO
GWO	1	The a = 2-2*(iter/$iter_{max}$)
GOA	1	c_{max} = 1, c_{min} = 0.00004 for finding value of c= c_{max}-l*((c_{max}- c_{min})/Max_iter, Max_iter = 500.
CSA	1	Awareness probability (AP) = 0.1, Flight length (FL) = 2.
ALO	2	As per paper [33] for ALO
ASO	1	The $\alpha = 50$ and $\beta = 0.2$.
TLBO	2	There is no such parameter to set value
SGO	2	There is only one parameter C called a self-introspection factor. Value of C is 0.2
GSA	1	G_0 = 100 and a = 20
LSA	2	Maximum channel time 10, energy equation based on Matlab code
DA	1	W = 0.9-0.2, s = 0.1, a = 0.1, c = 0.7, f = 1, e = 1
MSGO	2	C = 0.2. SAP = 0.7.

simulation runs with different randomly generated populations. Out of 32 runs, we have selected the best 30 runs. Statistical results on the best 30 in terms of best value(B), mean value(M) and corresponding standard deviation(SD) are reported in Tables 2–4. Results of Unimodal benchmark functions are given in Table 2, for multimodal benchmark functions given in Table 3 and for fixed-dimensional multimodal benchmark functions in Table 4. For every benchmark function, the best results are boldfaced. Results of the WRS test at a 0.05 significance level is conducted on experimental results, and finally, results are summarized.

Discussion on Test 1

According to Table 2, MSGO has gained the best performance, and it has reached to first rank among all algorithms. The Table shows that MSGO has consistently performed better than other algorithms. MSGO has an excellent performance in exploitation and convergence, and it successfully overcomes to solve all the problems within this category. It is clear from results that only SGO and MSGO achieve success in finding global optimum on F1 and F3 within 15,000 max_FEs. For F1, F2, F3, F4, F5, F7, the performance of MSGO is far better than other methods except for SGO and HHO. SGO has an equivalent result with MSGO for F1, F3, and F7. HHO has better results than MSGO for F5. Whereas for F 6, the CSA algorithm has the best result than among all algorithms. Similarly, DE, SSA, HHO, and GOA algorithm perform better than the MSGO algorithm for F6. From Table 2 and the WRS test, we conclude that the MSGO algorithm is superior to GA, PSO, ABC, WOA, MVO, SCA, LAPO, GWO, GSA, ALO, ASO, TLBO, LSA and DA algorithm in all seven unimodal benchmark functions. Whereas MSGO is better from the DE algorithm in 6 cases, and in 1 case, it is equivalent. Compared to SGO, MSGO is better in 4 cases better, and in 3 cases, it is equivalent. Similarly, when compared with CSA, it is better in 6 cases. While comparing with HHO, we can find that SGO is performing better only in 5 cases. And MSGO is better in 6 cases and 7 cases when compared with GOA and SSA, respectively. There are only seven unimodal benchmark functions reported in Table 2, and it can be observed that in the majority of cases, MSGO either performs equally good or outperforms the algorithms.

The multimodal functions test suits, i.e., functions F8–F13, are beneficial during the exploration capability of the optimization algorithm. Form Table 3, results show that the MSGO algorithm is eligible for solving problems with challenging search space. In this case, MSGO has demonstrated excellent performance in comparison, and it has reached to first rank among all algorithms. The table shows that MSGO has consistently performed better than other algorithms. MSGO has an excellent performance in exploration and convergence, and it successfully overcomes to solve all the problems within this category. It is clear from tabular results that only MSGO, WOA, HHO, TLBO, and SGO achieve success in finding global optimum on F9 and F11 within 15,000 max_FEs. For function F10, MSGO, WOA, HHO, TLBO, and SGO find equivalent results. Whereas for function F12, HHO has the best result than among all algorithms, and it is competitive with MSGO as well. GOA perform equivalent result with MSGO algorithm for F12, and HHO achieves equivalent result with MSGO for F13. Also, from Table 3 and the WRS test, we conclude that the MSGO algorithm is superior to GA, PSO, DE ABC, SSA, MVO, SCA, LAPO, GWO, CSA, ALO, ASO, GSA, LSA and DA algorithm in all six multimodal benchmark functions whereas it is performing better in 3cases, 1 case, 5 cases, 3 cases, and 3 cases in comparison to WOA, HHO, GOA, TLBO, and SGO respectively. From this test, we can confidently conclude that MSGO is far superior in solving multimodal benchmark functions in comparison to all other 20 algorithms studied in this work.

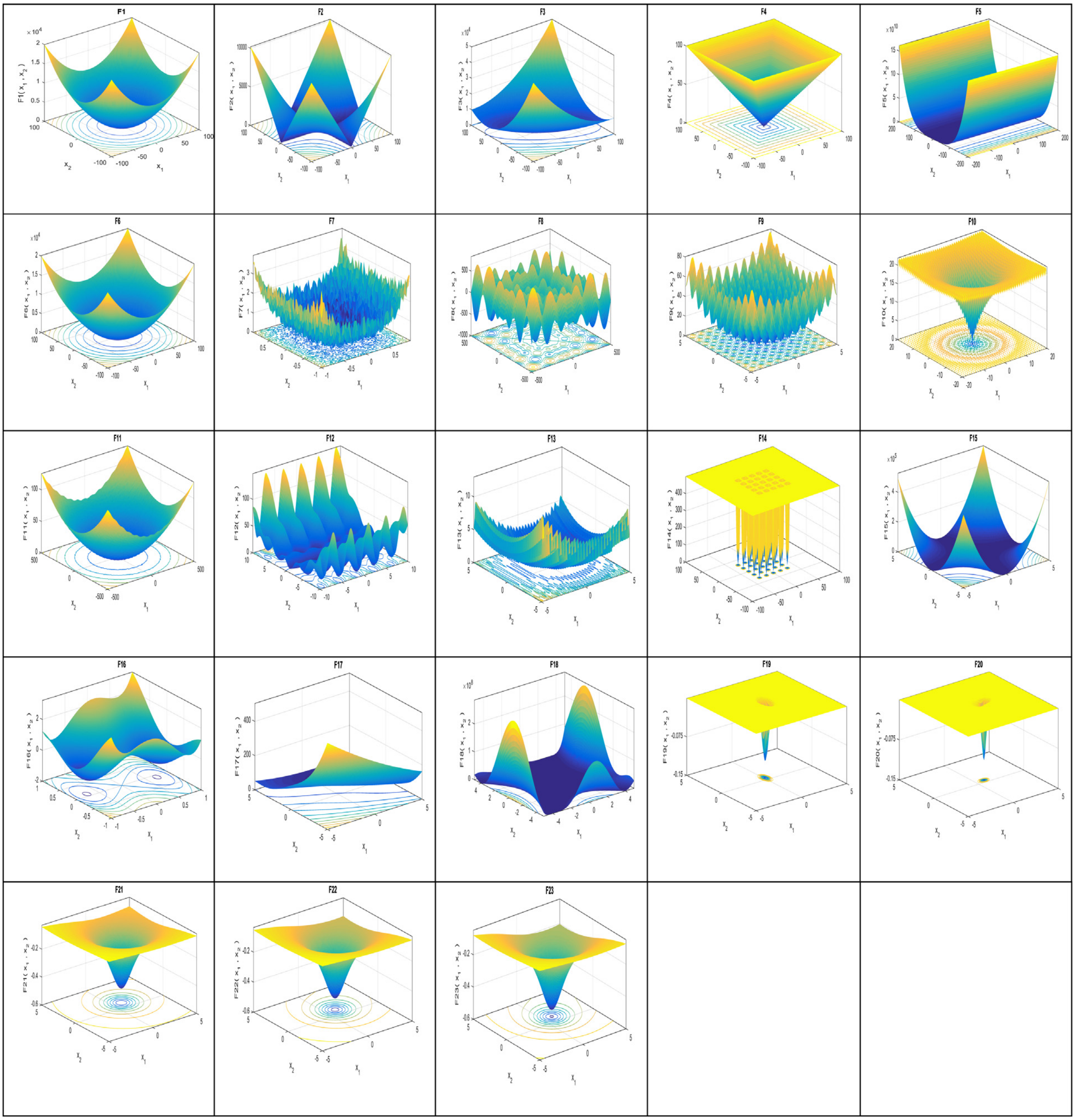

Fig. 2. The 3-D version of classical benchmark functions.

The fixed-dimensional multimodal functions are designed to have many local optimal points where computation complexity increases drastically with the problem size. The results reported from Table 4 for functions F14–F23 indicate that MSGO has an excellent exploration capability. The table shows that MSGO has consistently performed better than other algorithms. It is evident from results that MSGO algorithm achieves success in finding global optimum on F14, F16, F17, F18, F19, F20, F21, F22, F23 within 15,000 max_FEs.. and for the function F15, it almost reaches to global optimum (best value reach to global optimum, but mean value reach near-global optimum). From the observation of the WRS test reported in Table 4, we conclude that the MSGO algorithm either performs better or equivalent in comparison to all algorithms. It never shows worse than any 19 algorithms except GA. The results for GA is not computed since the application of mutation and crossover operator on a fixed dimension is not feasible.

From the Test 1 and analysis of Tables 2–4, respectively, the performance of MSGO is clearly evident. MSGO can provide the best results with better mean and standard deviations in the majority of benchmark functions in comparison to all 20 algorithms which are studied in our work.

Test 2

A deeper analysis of Test 1 results reveal that out of 20 algorithms, only four algorithms such as TLBO, SGO, WOA, and HHO can converge faster to locate global optima for unimodal and multimodal benchmark functions. This can be verified from the fact that numbers of FEs are less in comparison to the other 16 algorithms to reach global optimal value. Test 2 is aimed at experimenting with the capabilities of these algorithms to solve high dimensional problems and their performance comparisons with MSGO. Only 13 benchmark multimodal and unimodal functions with high dimensions starting with 100, 500, 1000, 1500,

Table 2
Results on unimodal functions on 30 repetitions in terms of Best (B), Mean(M), and Standard deviation (SD).

Functions		GA		PSO		DE		ABC		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F1	B	4.5931e+03		23.2860		2.6201e−12		5.1833		**0**
	M	8.9000e+03	−	76.9335	−	1.8812e−04	−	14.5733	−	**0**
	SD	1.8705e+03		31.9150		7.0503e−04		7.5363		**0**
F2	B	27.1073		2.3918		1.6397e−07		0.7384		**8.3178e−175**
	M	36.9301	−	7.3666	−	6.1642e−04	−	1.1845	−	**8.1015e−174**
	SD	3.4926		3.0318		0.0021		0.2592		**0**
F3	B	1.8293e+04		354.3102		137.5353		1.6298e+04		**0**
	M	2.6497e+04	−	2.6799e3	−	757.1439	−	2.5815e+04	−	**0**
	SD	3.8274e+03		2.6157e3		965.7426		3.6430e+03		**0**
F4	B	39.2518		4.3508		14.1436		46.7749		**4.1437e−174**
	M	46.3640	−	7.5786	−	25.9679	−	57.2323	−	**5.9665e−174**
	SD	3.6362		1.7179		6.0893		3.9239		**0**
F5	B	3.7540e+06		160.6205		20.2093		1.8295e+03		**5.6284e−04**
	M	6.5835e+06	−	1.4821e+03	−	334.0200	−	5.5065e+03	−	**0.0384**
	SD	2.0055e+06		1.7592e+03		444.3179		3.5868e+03		**0.0384**
F6	B	6331		26.2675		**5.0067e−05**		3.1398		4.2621e−04
	M	9.2558e+03	−	88.1072	−	**1.4954e−04**	+	18.9567	−	0.0018
	SD	1.6365e+03		68.0237		**4.4487e−04**		10.4361		0.0015
F7	B	2.5134		0.0038		0.0209		0.3353		**3.7585e−06**
	M	4.2536	−	0.0355	−	0.0364	−	0.4881	−	**7.6776e−05**
	SD	0.8049		0.0240		0.0129		0.0778		**3.8462e−05**
Total "−"			07		07		06		07	
Total "+"			00		00		01		00	
Total "≈"			00		00		00		00	
Functions		WOA		SSA		MVO		SCA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F1	B	2.5790e−87		2.2132e−08		0.6545		0.0229		**0**
	M	4.5919e−75	−	1.2865e−07	−	1.3765	−	5.3712	−	**0**
	SD	1.7944e−74		1.1112e−07		0.4151		7.5067		**0**
F2	B	4.8032e−58		0.0408		0.3876		1.1187e−04		**8.3178e−175**
	M	5.1593e−52	−	1.6646	−	1.0474	−	0.0109	−	**8.1015e−174**
	SD	1.3076e−51		0.9739		0.8877		0.0104		**0**
F3	B	1.6597e+04		290.2407		96.5933		1.5735e+03		**0**
	M	4.8524e+04	−	1.7025e+03	−	233.3219	−	8.6926e+03	−	**0**
	SD	1.4327e+04		903.6838		73.7320		5.9402e+03		**0**
F4	B	0.4100		6.0022		0.7843		16.0691		**4.1437e−174**
	M	49.1132	−	11.1143	−	2.0002	−	33.3291	−	**5.9665e−174**
	SD	23.3126		2.4177		0.9216		8.2537		**0**
F5	B	27.0741		26.1270		37.9123		56.7883		**5.6284e−04**
	M	27.8771	−	169.9430	−	252.7154	−	3.5254e+04	−	**0.0384**
	SD	0.4403		246.9790		334.3576		8.5047e+04		**0.0384**
F6	B	0.0861		**2.2837e−08**		0.7516		4.7445		4.2621e−04
	M	0.3811	−	**1.6934e−07**	+	1.2965	−	11.0355	−	0.0018
	SD	0.1735		**1.7188e−07**		0.3174		8.6013		0.0015
F7	B	5.7822e−05		0.1038		0.0153		0.0020		**3.7585e−06**
	M	0.0026	−	0.1775	−	0.0366	−	0.0900	−	**7.6776e−05**
	SD	0.0025		0.0463		0.0132		0.0733		**3.8462e−05**
Total "−"			07		06		07		07	
Total "+"			00		01		00		00	
Total "≈"			00		00		00		00	
Functions		LAPO		HHO		GWO		GOA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F1	B	3.2980e−08		3.3235e−112		3.1667e−29		1.1113e−09		**0**
	M	5.9199e−07	−	1.7238e−100	−	9.9446e−28	−	1.2886e−08	−	**0**
	SD	7.3060e−07		4.0005e−100		1.6500e−27		8.1914e−09		**0**
F2	B	2.0441e−05		2.8924e−60		1.0638e−17		0.0013		**8.3178e−175**
	M	1.5633e−04	−	2.7262e−52	−	6.6199e−17	−	0.8169	−	**8.1015e−174**
	SD	1.0585e−04		9.3362e−52		3.5127e−17		0.7945		**0**
F3	B	0.0011		4.1188e−98		8.7405e−09		1.1925e−09		**0**
	M	0.0406	−	8.9734e−80	−	1.3902e−05	−	8.1906e−07	−	**0**
	SD	0.0285		4.4589e−79		4.3624e−05		1.0063e−06		**0**

(*continued on next page*)

and 2000 dimensions are chosen as given in Appendix A for the purpose. In this test also the same approach of experimentation followed as in test 1 to generate results for comparison. A total of 32 simulations runs are done, and only the best 30 runs are

Table 2 (*continued*).

F4	B	1.7523e−04		2.8568e−57		9.9743e−08		3.5016e−05		**4.1437e−174**
	M	5.1081e−04	−	9.5811e−51	−	7.8728e−07	−	1.3535e−04	−	**5.9665e−174**
	SD	2.0533e−04		2.3635e−50		7.8828e−07		1.1532e−04		**0**
F5	B	25.1947		**4.6282e−05**		26.1129		0.0778		5.6284e−04
	M	26.2727	−	**0.0102**	+	27.0878	−	44.9977	−	0.0384
	SD	0.4690		**0.0110**		0.7116		104.1635		0.0384
F6	B	0.0085		7.0707e−07		0.2201		**2.0836e−09**		4.2621e−04
	M	0.0478	−	1.2130e−04	+	0.8153	−	**1.6022e−08**	+	0.0018
	SD	0.0221		1.3307e−04		0.3041		**1.0342e−08**		0.0015
F7	B	3.4707e−04		1.0447e−05		3.2372e−04		1.3673e−04		**3.7585e−06**
	M	0.0016	−	1.5171e−04	−	0.0018	−	0.0519	−	**7.6776e−05**
	SD	0.0010		1.1633e−04		9.0083e−04		0.0732		**3.8462e−05**
Total "−"			07		05		07		06	
Total "+"			00		02		00		01	
Total "≈"			00		00		00		00	

Functions		CSA		ALO		ASO		TLBO		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F1	B	3.1598e−18		5.0388		7.9215e−19		1.6014e−66		**0**
	M	1.4787e−17	−	203.3915	−	3.5249e−17	−	3.7067e−62	−	**0**
	SD	1.0130e−17		150.6267		5.1703e−17		1.0050e−61		**0**
F2	B	0.0270		5.8769		3.4188e−08		2.5608e−33		**8.3178e−175**
	M	0.8911	−	73.4102	−	0.0029	−	1.7704e−31	−	**8.1015e−174**
	SD	0.6086		44.0537		0.0099		3.1869e−31		**0**
F3	B	5.7667e−08		1.9376e+03		957.5297		2.5725e−65		**0**
	M	3.9433e−07	−	9.0974e+03	−	2.4017e+03	−	1.5185e−61	−	**0**
	SD	2.9388e−07		2.9960e+03		759.9022		2.7643e−61		**0**
F4	B	0.0045		15.2813		0.0024		1.1636e−32		**4.1437e−174**
	M	0.1443	−	21.3226	−	0.4751	−	2.2032e−30	−	**5.9665e−174**
	SD	0.1540		3.5050		0.5383		3.9999e−30		**0**
F5	B	20.6207		740.9816		25.5371		28.8366		**5.6284e−04**
	M	25.1459	−	1.3070e+04	−	54.4064	−	28.9332	−	**0.0384**
	SD	1.8076		1.0981e+04		60.8344		0.0336		**0.0384**
F6	B	**5.0851e−18**		5.5719		0		3.7102		4.2621e−04
	M	**1.1989e−17**	+	126.7143	−	0.6000	−	5.4567	−	0.0018
	SD	**5.1186e−18**		110.3013		0.7701		0.8195		0.0015
F7	B	0.0011		0.2454		0.0356		2.1972e−04		**3.7585e−06**
	M	0.0042	−	0.5417	−	0.0841	−	9.0450e−04	−	**7.6776e−05**
	SD	0.0013		0.1710		0.0277		4.3271e−04		**3.8462e−05**
Total "−"			06		07		07		07	
Total "+"			01		00		00		00	
Total "≈"			00		00		00		00	

Function		SGO		GSA		LSA		DA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F1	B	**0**		1.0098e−16		3.1233e−10		1.9392e−04		**0**
	M	**0**	≈	2.2679e−16	−	6.4463e−07	−	0.7027	−	**0**
	SD	**0**		7.8615e−17		1.3759e−06		1.1263		**0**
F2	B	1.8835e−173		6.1538e−08		0.0030		0.6350		**8.3178e−175**
	M	2.1921e−170	−	0.0436	−	0.1012	−	1.4578	−	**8.1015e−174**
	SD	0		0.1545		0.1407		0.8909		**0**
F3	B	**0**		330.0744		65.1953		0.6574		**0**
	M	**0**	≈	884.6245	−	111.2623	−	52.3336	−	**0**
	SD	**0**		304.1669		52.4058		35.3255		**0**
F4	B	1.3618e−173		2.9783		1.5840		0.2585		**4.1437e−174**
	M	1.3618e−170	−	7.2054	−	3.6624	−	2.1020	−	**5.9665e−174**
	SD	0		2.1624		1.8829		1.8413		**0**
F5	B	25.5980		26.1983		17.4627		7.9609		**5.6284e−04**
	M	26.5292	−	56.9628	−	60.7995	−	108.5300	−	**0.0384**
	SD	0.4422		43.6279		41.2541		179.5707		**0.0384**
F6	B	3.5512e−06		0		0		0.3470		**4.2621e−04**
	M	0.0144	−	5.2667	−	2.8000	−	2.0270	−	**0.0018**
	SD	0.0345		7.4368		2.9496		2.6023		**0.0015**
F7	B	**1.2047e−06**		0.0377		0.0129		0.0063		**3.7585e−06**
	M	**9.9617e−05**	≈	0.0738	−	0.0283	−	0.0289	−	**7.6776e−05**
	SD	**1.0782e−04**		0.0204		0.0086		0.0188		**3.8462e−05**
Total "−"			04		07		07		07	
Total "+."			00		00		00		00	
Total "≈"			03		00		00		00	

" −", "+ " and "≈" denote that performance of algorithms are worse, better and similar to MSGO respectively.

Table 3
Results on Multimodal Functions on 30 repetitions in terms of Best (B), Mean (M), and Standard deviation (SD).

Functions		GA		PSO		DE		ABC		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F8	B	−9.1398e+03		−9.3168e+3		−1.104e+04		−5.421e+03		**−1.2569e+04**
	M	−8.4702e+03	−	−7.771e+03	−	−9.916e+03	−	−5.037e+03	−	**−1.2569e+04**
	SD	357.8712		1.0051e+03		690.4544		168.1494		**1.7813e−010**
F9	B	132.3482		44.6941		18.9042		24.3595		**0**
	M	158.5735	−	84.6695	−	37.6106	−	30.9360	−	**0**
	SD	12.5460		22.4398		11.9314		4.8363		**0**
F10	B	13.7612		2.8692		0.9313		3.4924		**8.8818e−16**
	M	14.8680	−	4.6089	−	2.6724	−	4.3770	−	**8.8818e−16**
	SD	0.5653		1.0713		1.1995		0.5014		**0**
F11	B	58.0686		1.0935		2.3758e−04		1.0335		**0**
	M	87.1366	−	1.6135	−	0.0491	−	1.1260	−	**0**
	SD	13.0722		0.4053		0.0595		0.0567		**0**
F12	B	5.0715e+03		0.6399		0.0038		0.0582		**6.3342e−07**
	M	2.1859e+06	−	3.1318	−	3.4061	−	0.3590	−	**4.4519e−05**
	SD	1.2690e+06		1.7733		3.1716		0.1777		**3.5883e−05**
F13	B	2.9286e+06		2.3482		3.1951e−04		0.3822		**5.9574e−06**
	M	1.3513e+07	−	14.4184	−	11.7909	−	1.2131	−	**9.1201e−05**
	SD	5.1210e+06		11.1284		10.4224		0.6178		**9.4790e−05**
Total "−"			06		06		06		06	
Total "+"			00		00		00		00	
Total "≈"			00		00		00		00	

Functions		WOA		SSA		MVO		SCA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F8	B	−1.2569e+04		−8.7330e+03		−9.5246e+03		−4.5091e+03		**−1.2569e+04**
	M	−1.0329e+04	−	−7.6191e+03	−	−7.6980e+03	−	−3.7873e+03	−	**−1.2569e+04**
	SD	1.7639e+03		620.7100		750.4638		256.0681		**1.7813e−010**
F9	B	**0**		28.8538		69.4254		0.0273		**0**
	M	**0**	≈	55.1206	−	121.4520	−	33.8986	−	**0**
	SD	**0**		14.5282		0.5861		34.7343		**0**
F10	B	**8.8818e−16**		0.4204		0.7975		0.0109		**8.8818e−16**
	M	**4.2040e−15**	≈	2.5643	−	1.9776	−	13.9556	−	**8.8818e−16**
	SD	**2.4567e−15**		0.7591		0.5861		8.7368		**0**
F11	B	**0**		0.0011		0.7077		0.0708		**0**
	M	**0**	≈	0.0184	−	0.8549	−	0.8435	−	**0**
	SD	**0**		0.0122		0.0815		0.3854		**0**
F12	B	0.0057		1.9968		0.1080		1.1258		**6.3342e−07**
	M	0.0193	−	6.7007	−	2.2056	−	5.8754e+04	−	**4.4519e−05**
	SD	0.0070		2.5716		1.3652		2.6093e+05		**3.5883e−05**
F13	B	0.1106		0.0365		0.0639		3.7558		**5.9574e−06**
	M	0.4672	−	15.2370	−	0.1652	−	1.3315e+05	−	**9.1201e−05**
	SD	0.2444		11.4361		0.0961		3.0488e+05		**9.4790e−05**
Total "−"			03		06		06		06	
Total "+"			00		00		00		00	
Total "≈"			03		00		00		00	

Functions		LAPO		HHO		GWO		GOA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F8	B	−5.5774e+03		−1.2569e+04		−7.4028e+03		−1.9765e+03		**−1.2569e+04**
	M	−4.8835e+03	−	−1.2569e+04	−	−6.1755e+03	−	−1.1631e+03	−	**−1.2569e+04**
	SD	322.6229		0.7973		641.8496		192.1072		**1.7813e−010**
F9	B	8.0056e−08		**0**		5.6843e−14		0.0927		**0**
	M	9.6210	−	**0**	≈	1.3929	−	8.0139	−	**0**
	SD	36.6884		**0**		2.4293		5.0936		**0**
F10	B	5.5555e−05		**8.8818e−16**		6.4837e−14		4.0927e−05		**8.8818e−16**
	M	1.6988e−04	−	**8.8818e−16**	≈	1.0238e−13	−	1.0001	−	**8.8818e−16**
	SD	9.6606e−05		**0**		1.4444e−14		0.9105		**0**
F11	B	3.9965e−08		**0**		0		0.0320		**0**
	M	1.2827e−06	−	**0**	≈	0.0015	−	0.1487	−	**0**
	SD	2.3661e−06		**0**		0.0040		0.0652		**0**
F12	B	1.4439e−04		**3.4570e−10**		0.0067		1.5661e−07		6.3342e−07
	M	9.0946e−04	−	**7.6085e−06**	+	0.0333	−	4.2630e−05	≈	4.4519e−05
	SD	7.0833e−04		**9.9570e−06**		0.0181		1.0414e−04		3.5883e−05

(*continued on next page*)

considered for output generation. Populations are randomly generated for different benchmark functions. Best Values (B), Mean values (M) and standard deviation (SD) are noted and reported in Table 5. To compare the performances of these four algorithms

Table 3 (*continued*).

F13	B	0.0040		**4.5301e−08**		0.4122		2.6690e−08		**5.9574e−06**
	M	0.1134	–	**6.5837e−05**	≈	0.6604	–	9.0822e−04	–	**9.1201e−05**
	SD	0.0906		**6.0244e−05**		0.1667		0.0029		**9.4790e−05**
Total "–"			06		01		06		05	
Total "+"			00		01		00		00	
Total "≈"			00		04		00		01	
Functions		CSA		ALO		ASO		TLBO		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F8	B	−8.0882e+03		−6.9822e+03		−8.2647e+03		−6.3281e+03		**−1.2569e+04**
	M	−7.0164e+03	–	−5.5103e+03	–	−7.0546e+03	–	−5.0225e+03	–	**−1.2569e+04**
	SD	718.7470		283.3183		628.6317		522.5929		**1.7813e−010**
F9	B	12.9345		38.4101		21.8891		**0**		**0**
	M	29.2849	–	78.6814	–	35.7853	–	**0**	≈	**0**
	SD	13.3121		18.0437		7.8538		**0**		**0**
F10	B	2.3168		6.5805		7.1479e−10		**8.8818e−16**		**8.8818e−16**
	M	3.8464	–	11.4236	–	0.0195	–	**4.3225e−15**	≈	**8.8818e−16**
	SD	0.9613		1.9195		0.1039		**6.4863e−16**		**0**
F11	B	3.5194e−14		0.9815		0		**0**		**0**
	M	0.0101	–	2.4544	–	0.0021	–	**0**	≈	**0**
	SD	0.0118		1.3231		0.0041		**0**		**0**
F12	B	8.9308e−15		8.9731		1.0222e−20		0.3778		**6.3342e−07**
	M	1.6260	–	21.1000	–	0.0281	–	0.7056	–	**4.4519e−05**
	SD	1.7231		8.2490		0.0670		0.1968		**3.5883e−05**
F13	B	3.9301e−16		32.2766		2.3432e−19		1.8932		**5.9574e−06**
	M	0.0332	–	159.2096	–	0.0055	–	2.7348	–	**9.1201e−05**
	SD	0.0442		184.3326		0.0179		0.3558		**9.4790e−05**
Total "–"			06		06		06		03	
Total "+"			00		00		00		00	
Total "≈"			00		00		00		03	
Function		SGO		GSA		LSA		DA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F8	B	−1.0553e+04		−3.7138e+03		−8.5029e+03		−3.1083e+03		**−1.2569e+04**
	M	−7.8748e+03	–	−2.5180e+03	–	−7.6931e+03	–	−3.0362e+03	–	**−1.2569e+04**
	SD	1.4184e+03		318.7371		508.0486		86.5051		**1.7813e−010**
F9	B	**0**		19.8992		55.7179		10.2181		**0**
	M	**0**	≈	31.5402	–	64.6725	–	25.5718	–	**0**
	SD	**0**		6.5850		64.6725		10.7516		**0**
F10	B	**8.8818e−16**		7.3144e−09		1.5017		1.2001		**8.8818e−16**
	M	**8.8818e−16**	≈	1.2076e−08	–	2.4885	–	1.7211	–	**8.8818e−16**
	SD	**0**		2.9220e−09		0.8693		0.4330		**0**
F11	B	**0**		11.3759		5.0004e−08		0.2449		**0**
	M	**0**	≈	26.0273	–	0.0030	–	0.4380	–	**0**
	SD	**0**		5.3149		0.0066		0.1924		**0**
F12	B	4.5559e−07		0.1508		9.4165e−05		0.7934		**6.3342e−07**
	M	4.9003e−04	–	1.9442	–	0.3405	–	1.2777	–	**4.4519e−05**
	SD	0.0013		0.9927		0.4179		0.3263		**3.5883e−05**
F13	B	2.6036e−06		0.0339		1.5939e−04		0.0996		**5.9574e−06**
	M	0.0271	–	7.3968	–	0.0270	–	0.3278	–	**9.1201e−05**
	SD	0.0441		5.5807		0.0280		0.1693		**9.4790e−05**
Total "–"			03		06		06		06	
Total "+"			00		00		00		00	
Total "≈"			03		00		00		00	

" –", "+ " and "≈" denote that performance of algorithms are worse, better and similar to MSGO respectively.

with MSGO, the WRS test at a 0.05 significance level is conducted, and finally, results are summarized.

Discussion on test 2

Here seven unimodal and six multimodal benchmark functions of test1 of experiment 1 are considered for judging high dimensional parameter optimizations among WOA, TLBO, HHO, SGO, and MSGO algorithm by considering 100, 500, 1000, 1500,2000 dimensions. From Table 5, it is clear that MSGO and SGO achieve success in finding global optimum on F1, F2, F3, F4, F9, and F11 for all dimensions. WOA, TLBO, and HHO also achieve success in finding global optimum on F9 and F11. All algorithms find equivalent results on F10 for all dimensions. HHO algorithm finds better results than all other algorithms on F5, F6, F8. F12, and F13 for all dimensions.

From Table 5, we conclude that the performance of none of the algorithms, including MSGO is inferior while dealing with high dimensions. However, there is a relative difference among them while dealing with different dimensions. For example, for 100 and 500 dimensions, MSGO found to be superior to WOA in 10 cases and equivalent in 3 cases. The same can be seen when compared to TLBO. But when we compare the MSGO against its original version SGO, we find that MSGO produces better results in 5 cases, but both are equal in 8 cases. HHO algorithm fares well in comparison to MSGO since it is better in 4 cases and equivalence to MSGO in 4 cases. When the dimensions are further increased to 1000, 1500, and 2000, MSGO out classes WOA and TLBO since, in 10 cases, each it is better. This is also evident from Table 5 that MSGO produces better results in these dimensions

Table 4
Results on Fixed Dimensional Multimodal Benchmark Functions on 30 repetitions in terms of Best (B), Mean (M), and Standard Deviation (SD).

Functions				PSO		DE		ABC		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	**Results**
F14	B	NC		0.9980		0.9980		**0.9980**		**0.9980**
	M	NC	NC	1.0311	–	1.4568	–	**0.9980**	≈	**0.9980**
	SD	NC		0.1784		1.6536		**4.0101E-8**		**0**
F15	B	NC		3.0750E−4		3.0749E−4		8.0855E−4		**3.0750e−04**
	M	NC	NC	0.0056	–	0.0060	–	0.0017	–	**3.1167e−04**
	SD	NC		0.0084		0.0087		8.9604E−4		**8.7927e−06**
F16	B	NC		**- 1.0316**		**- 1.0316**		−1.0316		−**1.0316**
	M	NC	NC	**- 1.0316**	≈	**- 1.0316**	≈	2.2037E−8	–	−**1.0316**
	SD	NC		**2.7133E-11**		**5.5880E-16**		0.0706		**0**
F17	B	NC		**0.3979**		**0.3979**		**0.3979**		**0.3979**
	M	NC	NC	**0.3979**	≈	**0.3979**	≈	**0.3979**	≈	**0.3979**
	SD	NC		**6.1259E-12**		**0**		**2.9441E-5**		**0**
F18	B	NC		3.0000		**3.0000**		**3.0000**		**3.0000**
	M	NC	NC	5.7000	–	**3.0000**	≈	**3.0061**	≈	**3.0000**
	SD	NC		14.5399		**1.7123E-15**		**0.0078**		**3.8459e−16**
F19	B	NC		**- 3.8628**		**- 3.8628**		**- 3.8628**		−**3.8628**
	M	NC	NC	**- 3.8625**	≈	**- 3.8628**	≈	**- 3.8628**	≈	−**3.8628**
	SD	NC		**0.0014**		**2.5252E-15**		**1.6310E-5**		**2.7101e−15**
F20	B	NC		−3.3220		−3.3220		**- 3.3219**		−**3.3224**
	M	NC	NC	−3.2755	–	−3.2625	–	**- 3.3211**	≈	−**3.3219**
	SD	NC		0.0681		0.0594		**8.4270E-4**		**0.0051**
F21	B	NC		−10.1532		−10.1532		−10.1476		−**10.1532**
	M	NC	NC	−6.9746	–	−5.1345	–	−10.0188	–	−**10.1532**
	SD	NC		3.4764		2.9569		0.0974		**8.8818e−16**
F22	B	NC		−10.4029		−10.4029		−10.4007		−**10.4029**
	M	NC	NC	−7.8672	–	−6.0078	–	−10.2614	–	−**10.4029**
	SD	NC		3.3898		3.4431		0.1857		**0**
F23	B	NC		−10.5364		−10.5364		−10.5074		−**10.5364**
	M	NC	NC	−7.0843	–	−7.2796	–	−10.1530	–	−**10.5364**
	SD	NC		3.8065		3.7594		0.4089		**1.2561e−15**
Total "−"			NC		07		06		05	
Total "+"			NC		00		00		00	
Total "≈"			NC		03		04		05	
Functions		WOA		SSA		MVO		SCA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F14	B	0.9980		0.9980		**0.9980**		0.9980		**0.9980**
	M	2.8004	–	1.1637	–	**0.9980**	≈	1.5934	–	**0.9980**
	SD	2.9381		0.3768		**3.8424e−11**		0.9247		**0**
F15	B	**3.1223e−04**		3.2831e−04		5.6302e−04		**4.8095e−04**		**3.0750e−04**
	M	**7.0984e−04**	≈	0.0015	–	0.0081	–	**9.9318e−04**	≈	**3.1167e−04**
	SD	**3.9093e−04**		0.0036		0.0136		**3.2599e−04**		**8.7927e−06**
F16	B	−**1.0316**		−**1.0316**		−**1.0316**		−**1.0316**		−**1.0316**
	M	−**1.0316**	≈	−**1.0316**	≈	−**1.0316**	≈	−**1.0316**	≈	−**1.0316**
	SD	**3.1248e−10**		**2.2470e−14**		**4.0922e−07**		**3.9343e−05**		**0**
F17	B	**0.3979**		**0.3979**		**0.3979**		0.3979		**0.3979**
	M	**0.3979**	≈	**0.3979**	≈	**0.3979**	≈	0.3984	–	**0.3979**
	SD	**2.6146e−06**		**2.4512e−07**		**6.0267e−07**		0.0021		**0**
F18	B	**3.0000**		**3.0000**		**3.0000**		**3.0000**		**3.0000**
	M	**3.0000**	≈	**3.0000**	≈	**3.0000**	≈	**3.0001**	≈	**3.0000**
	SD	**5.1803e−05**		**1.8153e−13**		**2.7807e−06**		**6.0981e−05**		**3.8459e−16**
F19	B	−3.8628		−**3.8628**		−3.8628		−3.8612		−**3.8628**
	M	−3.8583	–	−**3.8628**	≈	−3.8628	–	−3.8545	–	−**3.8628**
	SD	0.0049		**1.7851e−11**		3.1204e−06		0.0020		**2.7101e−15**
F20	B	−3.3216		−3.3220		−3.3220		−3.1262		−**3.3224**
	M	−3.2363	–	−3.2424	–	−3.2575	–	−3.0150	–	−**3.3219**
	SD	0.0855		0.0618		0.0614		0.0907		**0.0051**
F21	B	−10.1505		−10.1532		−10.1532		−4.9442		−**10.1532**
	M	−8.3343	–	−8.2299	–	−7.2107	–	−2.5997	–	−**10.1532**
	SD	2.5712		3.0601		3.1048		1.8994		**8.8818e−16**
F22	B	−10.4021		−10.4029		−10.4029		−6.7813		−**10.4029**
	M	−7.5655	–	−8.4447	–	−8.1956	–	−4.2889	–	−**10.4029**
	SD	2.9104		3.3283		3.0419		1.2965		**0**

(*continued on next page*)

Table 4 (*continued*).

F23	B	−10.5345	−	−10.5364	−	−10.5363	−	−8.9267	−	**−10.5364**
	M	−6.5336		−8.9911		−8.3881		−4.0692		**−10.5364**
	SD	3.1086		3.1516		3.1869		1.8162		**1.2561e−15**
Total "−"			06		06		06		07	
Total "+"			00		00		00		00	
Total "≈"			04		04		04		03	
Functions		LAPO		HHO		GWO		GOA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F14	B	**0.9980**	≈	0.9980	−	0.9980	−	**0.9980**	≈	**0.9980**
	M	**0.9980**		1.2618		4.7796		**0.9980**		**0.9980**
	SD	**4.4511e−08**		0.9320		4.5220		**5.2723e−16**		**0**
F15	B	**3.1289e−04**	≈	**3.0790e−04**	≈	3.0753e−04	−	4.1043e−04	−	**3.0750e−04**
	M	**5.7812e−04**		**3.4235e−04**		0.0038		0.0037		**3.1167e−04**
	SD	**2.5619e−04**		**3.3819e−05**		0.0075		0.0068		**8.7927e−06**
F16	B	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**
	M	**−1.0316**		**−1.0316**		**−1.0316**		**−1.0316**		**−1.0316**
	SD	**2.9012e−07**		**2.5853e−10**		**1.8691e−08**		**1.6029e−13**		**0**
F17	B	0.3979	−	**0.3979**	≈	**0.3979**	≈	**0.3979**	≈	**0.3979**
	M	0.3983		**0.3979**		**0.3979**		**0.3979**		**0.3979**
	SD	6.8423E−4		**3.2708e−06**		**1.3408e−06**		**3.8926e−13**		**0**
F18	B	**3.0000**	≈	**3.0000**	≈	**3.0000**	≈	**3.0000**	≈	**3.0000**
	M	**3.0001**		**3.0000**		**3.0000**		**3.0000**		**3.0000**
	SD	**3.2314e−16**		**6.5683e−08**		**2.7401e−05**		**2.0002e−12**		**3.8459e−16**
F19	B	**−3.8628**	≈	−3.8628	−	**−3.8628**	≈	−3.8628	−	**−3.8628**
	M	**−3.8628**		−3.8607		**−3.8621**		−3.8490		**−3.8628**
	SD	**2.6712e−16**		0.0021		**0.0014**		0.0276		**2.7101e−15**
F20	B	3.3220	−	−3.2705	−	−3.3220	−	−3.3220	−	**−3.3224**
	M	−3.2729		−3.1115		−3.2537		−3.2695		**−3.3219**
	SD	0.0571		0.0803		0.0718		0.0610		**0.0051**
F21	B	−10.1532	−	−5.0549	−	−10.1528	−	−10.1532	−	**−10.1532**
	M	−9.6960		−5.0519		−9.4777		−4.6385		**−10.1532**
	SD	0.8042		0.0031		1.7467		2.4744		**8.8818e−16**
F22	B	−10.4029	−	−10.3236	−	−10.4025	−	−10.4029	−	**−10.4029**
	M	−10.1728		−5.5891		−10.4010		−6.9588		**−10.4029**
	SD	0.6905		1.5436		8.9614e−04		3.5779		**0**
F23	B	−10.5364	−	−5.1281	−	−10.5363	−	−10.5364	−	**−10.5364**
	M	−10.2295		−5.1241		−10.5346		−5.3619		**−10.5364**
	SD	0.6352		0.0043		9.5692e−04		3.5483		**1.2561e−15**
Total "−"			05		06		06		06	
Total "+"			00		00		00		00	
Total "≈"			05		04		04		04	
Functions		CSA		ALO		ASO		TLBO		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F14	B	**0.9980**	≈	0.9980	−	0.9980	−	**0.9980**	≈	**0.9980**
	M	**0.9980**		3.3020		1.3293		**0.9980**		**0.9980**
	SD	**0**		2.1828		0.5429		**4.1233e−17**		**0**
F15	B	**3.0749e−04**	≈	4.6283e−04	−	**6.1340e−04**	≈	**3.0749e−04**	≈	**3.0750e−04**
	M	**3.3801e−04**		0.0018		**9.1801e−04**		**4.3503e−04**		**3.1167e−04**
	SD	**1.6718e−04**		0.0031		**1.8598e−04**		**1.9521e−04**		**8.7927e−06**
F16	B	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**
	M	**−1.0316**		**−1.0316**		**−1.0316**		**−1.0316**		**−1.0316**
	SD	**6.7752e−16**		**1.0090e−13**		**6.5195e−16**		**6.7752e−16**		**0**
F17	B	**0.3979**	≈	**0.3979**	≈	**0.3979**	≈	**0.3979**	≈	**0.3979**
	M	**0.3979**		**0.3979**		**0.3979**		**0.3979**		**0.3979**
	SD	**2.6978e−06**		**1.5874e−13**		**0**				**0**
F18	B	**3.0000**	≈	**3.0000**	≈	**3.0000**	≈	**3.0000**	≈	**3.0000**
	M	**3.0000**		**3.0000**		**3.0000**		**3.0000**		**3.0000**
	SD	**2.1709e−15**		**8.9882e−13**		**2.4740e−15**		**1.7324e−15**		**3.8459e−16**
F19	B	**−3.8628**	≈	**−3.8628**	≈	**−3.8628**	≈	**−3.8628**	≈	**−3.8628**
	M	**−3.8628**		**−3.8628**		**−3.8628**		**−3.8628**		**−3.8628**
	SD	**2.7101e−15**		**1.4133e−11**		**2.5973e−15**		**2.6576e−09**		**2.7101e−15**
F20	B	−3.3220	−	−3.3220	−	**−3.3220**	≈	−3.3218	−	**−3.3224**
	M	−3.3101		−3.3016		**−3.3220**		−3.2967		**−3.3219**
	SD	0.0363		0.0464		**1.3424e−15**		0.0461		**0.0051**
F21	B	**−10.1532**	≈	−10.1532	−	−10.1532	−	−10.1532	−	**−10.1532**
	M	**−10.1532**		−5.7704		−8.2508		−9.7561		**−10.1532**
	SD	**7.2269e−15**		2.3735		3.1179		1.0122		**8.8818e−16**

(*continued on next page*)

Table 4 (*continued*).

	B	**−10.4029**		−10.4029		**−10.4029**		−10.4029		**−10.4029**
F22	M	**−10.4029**	≈	−6.8464	−	**−10.4029**	≈	−9.6836	−	**−10.4029**
	SD	**1.7140e−15**		3.4643		**5.7134e−16**		1.7405		**0**
	B	**−10.5364**		−10.5364		−10.5364		−10.5364		**−10.5364**
F23	M	**−10.5364**	≈	−5.7568	−	−10.4156	−	−10.5249	−	**−10.5364**
	SD	**1.7764e−15**		3.2839		0.6617		0.0356		**1.2561e−15**
Total "−"			01		06		07		04	
Total "+"			00		00		00		00	
Total "≈"			09		04		03		06	
Function		SGO		GSA		LSA		DA		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
	B	0.9980		1.0056		**0.9980**		**0.9980**		**0.9980**
F14	M	2.4676	−	5.2491	−	**0.9980**	≈	**0.9980**	≈	**0.9980**
	SD	3.1852		2.9338		**0**		**1.1102e−16**		**0**
	B	**3.0749e−04**		9.1674e−04		**3.0763e−04**		3.4834e−04		**3.0750e−04**
F15	M	**3.2012e−04**	≈	0.0038	−	**4.4161e−04**	≈	0.0047	−	**3.1167e−04**
	SD	**3.1712e−05**		0.0017		**9.9419e−05**		0.0069		**8.7927e−06**
	B	**−1.0316**		**−1.0316**		**−1.0316**		**−1.0316**		**−1.0316**
F16	M	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**	≈	**−1.0316**
	SD	**6.7752e−16**		**5.2156e−16**		**0**		**1.9230e−16**		**0**
	B	**0.3979**		**0.3979**		**0.3979**		**0.3979**		**0.3979**
F17	M	**0.3979**	≈	**0.3979**	≈	**0.3979**	≈	**0.3979**	≈	**0.3979**
	SD	**0**		**0**		**0**		**0**		**0**
	B	**3.0000**		**3.0000**		**3.0000**		**3.0000**		**3.0000**
F18	M	**3.0000**	≈	**3.0000**	≈	**3.0000**	≈	**3.0000**	≈	**3.0000**
	SD	**1.4958e−15**		**3.3171e−15**		**7.6919e−16**		**3.5388e−15**		**3.8459e−16**
	B	**−3.8628**		**−3.8628**		**−3.8628**		**−3.8628**		**−3.8628**
F19	M	**−3.8628**	≈	**−3.8628**	≈	**−3.8628**	≈	**−3.8628**	≈	**−3.8628**
	SD	**2.6823e−15**		**2.3237e−15**		**0**		**1.9604e−06**		**2.7101e−15**
	B	−3.3220		**−3.3220**		**−3.3220**		**−3.3220**		**−3.3224**
F20	M	−3.3101	−	**−3.3220**	≈	**−3.3220**	≈	**−3.3177**	≈	**−3.3219**
	SD	0.0363		**1.6118e−15**		**0**		**0.0074**		**0.0051**
	B	−10.1532		−10.1532		**−10.1532**		−10.1532		**−10.1532**
F21	M	−7.0552	−	−6.3368	−	**−10.1532**	≈	−9.1411	−	**−10.1532**
	SD	9.0336e−03		3.6678		**1.2561e−15**		2.2586		**8.8818e−16**
	B	−10.4029		**−10.4029**		**−10.4029**		−10.4029		**−10.4029**
F22	M	−8.2648	−	**−10.4029**	≈	**−10.4029**	≈	−7.2325	−	**−10.4029**
	SD	0.9704		**9.3299e−16**		**0**		2.8806		**0**
	B	−10.5363		**−10.5364**		−10.5364		**−10.5364**		**−10.5364**
F23	M	−8.1285	−	**−10.5364**	≈	−5.1756	−	**−10.5364**	≈	**−10.5364**
	SD	3.7465e−04		**3.0232e−15**		2.9968		**9.9692e−06**		**1.2561e−15**
Total "−"			05		03		01		03	
Total "+"			00		00		00		00	
Total "≈"			05		07		09		07	

" −", "+ " and "≈" denote that performance of algorithms are worse, better and similar to MSGO respectively.

compared to SGO. However, HHO once again found to be resulting in better outcomes in these high dimensions compared to MSGO.

Hence, from the above analysis, we can fairly conclude that MSGO is a better meta-heuristic algorithm in dealing with high dimensional unimodal and multimodal benchmark functions.

Experiment 2

In this experiment, BOA (Butterfly Optimization Algorithm) [40] is compared with the MSGO algorithm. For comparison of the performance of algorithms, 30 classical benchmark functions are considered. These benchmark functions are described in [40]. We have derived BOA results from the paper [40].

For an experiment, common control parameters such as pop_size are set as 50, and maximum iterations are set as 10,000 for the BOA algorithm in its respective paper [40]. But for the MSGO algorithm, we have set as pop_size is 25, and the maximum iteration is 500. So max_FEs are set to 25,000 (225500 = 25,000, two is due to two-time fitness calculation in one iteration for one particle in population) for the MSGO algorithm.

For each function, algorithms run 30 times with different randomly generated populations. Statistical results in terms of mean value and corresponding standard-deviation, the best value, Median value, and worse value are reported in Table 6. For every benchmark function, the best results are boldfaced. WRS test at a 0.05 significance level is conducted on the experimental results, and the last three rows of Table 6 summarize the results also.

Discussion

From Table 6, it is clear that MSGO algorithm reaches global optimum for 22 benchmark functions such as F1–F4, F6, F8, F11–F13, F15–F24, F26, F27, F29 whereas BOA reaches to global optimum for 18 benchmark functions such as F1–F4, F6, F11–F13, F15–F17, F19, F21–F24, F27, F29. MSGO also shows its dominating performance in most functions and satisfactory results in 10 functions such as F7, F8, F9, F10, F14, F18, F20, F25, F26, F30. Particular attention should be paid to the noisy (Quartic) problems as these challenges frequently occur in real-world applications. MSGO provided a significant performance boost on these noisy problems and gave an equivalent solution in comparison to BOA, but its best result is better than BOA's best result. Besides optimization accuracy, convergence speed is quite essential to an optimizer. In this experiment, we use a maximum number of 500 iterations as the termination criterion, whereas for BOA maximum number of 10,000 iterations [40] as the termination criterion. In Table 6

Table 5
Results of high dimensional benchmark functions.

Functions		WOA Results	WOA WRS	TLBO Results	TLBO WRS	SGO Results	SGO WRS	HHO Results	HHO WRS	MSGO Results
Dim=100										
F1	B	6.8936e−88		4.5784e−138		**0**		1.4747e−115		**0**
	M	2.8215e−85	−	1.2698e−134	−	**0**	≈	1.2525e−104	−	**0**
	SD	4.9718e−85		2.5717e−134		**0**		2.8003e−104		**0**
F2	B	6.1678e−57		8.4719e−71		**0**		2.0030e−58		**0**
	M	5.9222e−55	−	3.2334e−69	−	**0**	≈	5.5363e−53	−	**0**
	SD	1.1352e−54		2.9130e−69		**0**		1.2374e−52		**0**
F3	B	5.7062e+05		9.3817e−139		**0**		1.4133e−94		**0**
	M	7.6543e+05	−	8.8734e−134	−	**0**	≈	1.8386e−87	−	**0**
	SD	1.4946e+05		1.9433e−133		**0**		3.7603e−87		**0**
F4	B	46.3532		1.1253e−67		**0**		8.7142e−57		**0**
	M	79.5968	−	1.0685e−66	−	**0**	≈	6.0277e−52	−	**0**
	SD	18.6633		9.7421e−67		**0**		1.3350e−51		**0**
F5	B	97.9049		98.9173		96.3243		0.0011		6.0273e−05
	M	98.0624	−	98.9382	−	96.6386	−	0.0044	+	0.0463
	SD	0.1116		0.0194		0.2004		0.0033		0.0374
F6	B	1.3382		22.0715		0.0686		1.2120e−05		9.2124e−04
	M	1.8255	−	22.9250	−	0.1159	−	1.0253e−04	+	0.0055
	SD	0.3329		0.5924		0.0546		6.1221e−05		0.0046
F7	B	9.5398e−05		8.0568e−05		**6.2172e−06**		**1.3422e−05**		**1.3972e−05**
	M	0.0011	−	1.7956e−04	−	**2.4376e−05**	≈	**4.2935e−05**	≈	**4.0459e−05**
	SD	0.0013		1.0205e−04		**1.5737e−05**		**3.2080e−05**		**2.4858e−05**
F8	B	−4.1890e+04		−1.0180e+04		−3.0033e+04		−4.1898e+04		−4.1898e+04
	M	−4.1157e+04	−	−9.3734e+03	−	−2.8139e+04	−	−4.1887e+04	−	−4.1898e+04
	SD	736.7037		715.0089		1.6363e+03		0.4825		0.0977
F9	B	**0**		**0**		**0**		**0**		**0**
	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
F10	B	**8.8818e−16**		4.4409e−15		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**
	M	1.5987e−15	≈	**4.4409e−15**	≈	**8.8818e−16**	≈	**8.8818e−16**	≈	**8.8818e−16**
	SD	1.5888e−15		**0**		**0**		**0**		**0**
F11	B	**0**		**0**		**0**		**0**		**0**
	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
F12	B	0.0114		0.8942		2.8391e−05		**2.1876e−07**		5.2540e−07
	M	0.0178	−	1.0034	−	0.0020	−	**1.0099e−06**	+	2.6160e−05
	SD	0.0038		0.0936		0.0020		**6.3529e−07**		2.5336e−05
F13	B	0.6259		9.9899		0.0134		**1.1839e−05**		1.7278e−05
	M	1.0000	−	9.9924	−	0.2582	−	**1.7067e−05**	+	4.6915e−04
	SD	0.3129		0.0022		0.3189		**7.3751e−06**		5.9366e−04
Total "−"			10		10		05		05	
Total "+"			00		00		00		04	
Total "≈"			03		03		08		04	
Dim=500										
F1	B	9.1913e−90		8.4002e−140		**0**		3.0988e−112		**0**
	M	1.5150e−84	−	1.3344e−135	−	**0**	≈	3.2907e−107	−	**0**
	SD	2.1682e−84		2.1660e−135		**0**		4.3772e−107		**0**
F2	B	1.4611e−56		1.4329e−70		**0**		7.2302e−59		**0**
	M	8.0480e−52	−	8.0093e−68	−	**0**	≈	7.4813e−55	−	**0**
	SD	1.2130e−51		1.4724e−67		**0**		6.9025e−55		**0**
F3	B	1.4211e+07		4.8032e−135		**0**		1.2905e−86		**0**
	M	2.0256e+07	−	3.9507e−132	−	**0**	≈	5.9554e−76	−	**0**
	SD	6.0638e+06		8.5831e−132		**0**		1.1126e−75		**0**
F4	B	69.0105		1.0052e−68		**0**		1.4687e−58		**0**
	M	77.3727	−	1.2546e−66	−	**0**	≈	1.3970e−55	−	**0**
	SD	7.9106		2.2878e−66		**0**		1.4831e−55		**0**
F6	B	495.0045		498.8863		493.8094		8.8331e−04		0.0024
	M	495.3095	−	498.9308	−	493.9379	−	**0.0433**	+	2.5757
	SD	0.1745		0.0313		0.1369		**0.0546**		3.4200
F6	B	9.6508		122.1612		1.4874		**1.1064e−04**		0.0020
	M	13.2224	−	122.8960	−	2.3131	−	**2.0014e−04**	+	0.0222
	SD	2.2789		0.4850		1.0019		**1.2819e−04**		0.0361

(*continued on next page*)

Table 5 (*continued*).

Functions		WOA		TLBO		SGO		HHO		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
	B	2.3326e−05		1.5620e−04		**9.0047e−06**		**3.2216e−06**		**1.1663e−05**
F7	M	4.5523e−04	−	4.6921e−04	−	**1.7128e−05**	≈	**2.5630e−05**	≈	**4.1701e−05**
	SD	3.9885e−04		1.9672e−04		**7.4144e−06**		**1.8311e−05**		**3.5596e−05**
	B	−2.0902e+05		−2.1623e+04		−1.0126e+05		−2.0949e+05		**−2.0949e+05**
F8	M	−1.9321e+05	−	−2.1053e+04	−	−8.4757e+04	−	−2.0241e+05	−	**−2.0949e+05**
	SD	2.4459e+04		514.1468		1.4853e+04		1.3834		**0.2388**
	B	**0**		**0**		**0**		**0**		**0**
F9	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
	B	**4.4409e−15**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**
F10	M	**4.4409e−15**	≈	**3.7303e−15**	≈	**8.8818e−16**	≈	**8.8818e−16**	≈	**8.8818e−16**
	SD	**0**		**1.5888e−15**		**0**		**0**		**0**
	B	**0**		**0**		**0**		**0**		**0**
F11	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
	B	0.0196		1.1233		0.0040		**1.2586e−08**		7.9103e−10
F12	M	0.0261	−	1.1539	−	0.0092	−	**1.7778e−07**	+	1.2050e−06
	SD	0.0052		0.0215		0.0069		**2.0798e−07**		2.1844e−06
	B	6.7739		49.9854		2.3923		**2.6445e−05**		2.8955e−05
F13	M	7.4445	−	49.9917	−	6.6416	−	**1.7500e−04**	+	0.0027
	SD	0.4366		0.0038		6.0345		**2.4534e−04**		0.0043
Total "−"			10		10		05		05	
Total "+"			00		00		00		04	
Total "≈"			03		03		08		04	
Dim=1000										
	B	2.7843e−87		2.4594e−137		**0**		9.6220e−120		**0**
F1	M	6.0686e−85	−	4.4568e−134	−	**0**	≈	1.2032e−105	−	**0**
	SD	6.5360e−85		9.6983e−134		**0**		2.3050e−105		**0**
	B	1.9040e−56		1.7573e−68		**0**		2.8570e−57		**0**
F2	M	2.0892e−53	−	8.0223e−59	−	**0**	≈	1.0322e−53	−	**0**
	SD	2.7064e−53		1.7939e−57		**0**		2.0084e−53		**0**
	B	5.7480e+07		3.2900e−137		**0**		1.3436e−79		**0**
F3	M	1.0632e+08	−	2.5580e−131	−	**0**	≈	8.1618e−52	−	**0**
	SD	2.7699e+07		3.3075e−131		**0**		1.8250e−51		**0**
	B	22.0622		1.2022e−69		**0**		5.7213e−56		**0**
F4	M	56.4083	−	3.3129e−67	−	**0**	≈	3.5663e−52	−	**0**
	SD	25.0058		3.9644e−67		**0**		5.2142e−52		**0**
	B	990.8468		998.9004		989.3952		**0.0040**		0.2056
F5	M	991.5860	−	998.9270	−	990.5227	−	**0.1560**	+	1.9627
	SD	0.7759		0.0246		1.2006		**0.1754**		2.0946
	B	27.3725		247.2772		1.7882		**9.1554e−06**		0.0093
F6	M	35.2519	−	248.1814	−	9.5576	−	**2.0157e−04**	+	0.0851
	SD	5.6045		0.8088		6.7626		**1.8051e−04**		0.1064
	B	4.0686e−04		2.2180e−04		**4.0824e−06**		**2.2481e−05**		**2.3454e−06**
F7	M	0.0011	−	2.8975e−04	−	**3.1715e−05**	≈	**5.0719e−05**	≈	**2.2565e−05**
	SD	6.1710e−04		5.5630e−05		**3.0574e−05**		**2.5828e−05**		**2.3539e−05**
	B	−4.1830e+05		−3.2917e+04		−2.2891e+05		**−4.1898e+05**		−4.1898e+05
F8	M	−4.0497e+05	−	−2.9329e+04	−	−1.7005e+05	−	**−4.1898e+05**	+	−4.1898e+05
	SD	1.1382e+04		3.5733e+03		3.3712e+04		**2.7115**		0.5312
	B	**0**		**0**		**0**		**0**		**0**
F9	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
	B	**8.8818e−16**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**
F10	M	**2.3093e−15**	≈	**3.0198e−15**	≈	**8.8818e−16**	≈	**8.8818e−16**	≈	**8.8818e−16**
	SD	**1.9459e−15**		**1.9459e−15**		**0**		**0**		**0**
	B	**0**		**0**		**0**		**0**		**0**
F11	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
	B	0.0160		1.1487		0.0033		**6.3254e−10**		5.5165e−08
F12	M	0.0346	−	1.1579	−	0.0116	−	**5.5918e−07**	+	1.6370e−06
	SD	0.0180		0.0098		0.0064		**6.1459e−07**		2.0107e−06
	B	14.7520		99.9814		2.8942		**5.2503e−06**		4.6152e−04
F13	M	17.4280	−	99.9895	−	8.9500	−	**8.8515e−05**	+	0.0025
	SD	2.5766		0.0052		4.9460		**8.4492e−05**		0.0025

(*continued on next page*)

Table 5 (*continued*).

Functions		WOA		TLBO		SGO		HHO		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
Total "−"			10		10		05		04	
Total "+"			00		00		00		05	
Total "≈"			03		03		08		04	
Dim=1500										
	B	4.8206e−89		1.2390e−138		**0**		8.6837e−111		**0**
F1	M	7.0400e−82	−	3.8493e−135	−	**0**	≈	3.1219e−106	−	**0**
	SD	1.5732e−81		6.3523e−135		**0**		6.7858e−106		**0**
	B	2.4835e−57		5.3962e−64		**0**		8.0179e−56		**0**
F2	M	1.0398e−52	−	2.7710e−47	−	**0**	≈	5.4413e−53	−	**0**
	SD	1.5571e−52		6.1861e−47		**0**		1.1568e−52		**0**
	B	1.4552e+08		5.0094e−135		**0**		1.3436e−75		**0**
F3	M	2.1181e+08	−	3.6295e−132	−	**0**	≈	8.1618e−49	−	**0**
	SD	3.7570e+07		5.5325e−132		**0**		1.8250e−49		**0**
	B	47.9089		2.2866e−69		**0**		6.2254e−55		**0**
F4	M	64.2824	−	3.7341e−67	−	**0**	≈	4.8292e−53	−	**0**
	SD	14.3006		5.3607e−67		**0**		6.0524e−53		**0**
	B	1.4881e+03		1.4989e+03		1.4838e+03		**0.0060**		0.0164
F5	M	1.4890e+03	−	1.4989e+03	−	1.4849e+03	−	**0.0863**	+	0.9976
	SD	0.7085		0.0170		0.6457		**0.0773**		0.8168
	B	37.1219		372.0495		6.2696		**1.5677e−04**		0.0044
F6	M	56.1256	−	373.5717	−	13.9194	−	**9.6564e−04**	+	0.1120
	SD	13.2733		0.8820		7.3690		**7.2973e−04**		0.1774
	B	1.9886e−04		9.1026e−05		**2.3583e−05**		**3.4929e−06**		**2.1312e−05**
F7	M	8.6411e−04	−	2.2124e−04	−	**5.4661e−05**	≈	**6.2140e−05**	≈	**4.8862e−05**
	SD	9.4224e−04		9.3921e−05		**4.1699e−05**		**4.3969e−05**		**2.9344e−05**
	B	−6.2831e+05		−3.8756e+04		−3.1246e+05		−6.2847e+05		−6.2847e+05
F8	M	−6.1921e+05	−	−3.6374e+04	−	−2.4071e+05	−	−6.2847e+05	+	−6.2847e+05
	SD	1.7868e+04		1.5507e+03		5.6520e+04		4.9658		0.4608
	B	**0**		**0**		**0**		**0**		**0**
F9	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
	B	**4.4409e−15**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**
F10	M	**5.1514e−15**	≈	**3.0198e−15**	≈	**8.8818e−16**	≈	**8.8818e−16**	≈	**8.8818e−16**
	SD	**1.5888e−15**		**1.9459e−15**		**0**		**0**		**0**
	B	**0**		**0**		**0**		**0**		**0**
F11	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
	B	0.0211		1.1729		0.0043		**8.3611e−08**		3.9692e−07
F12	M	0.0352	−	1.1754	−	0.0078	−	**2.4566e−07**	+	5.8438e−06
	SD	0.0104		0.0024		0.0046		**1.4253e−07**		7.6856e−06
	B	18.3687		149.9788		18.1095		**1.0790e−06**		2.9251e−04
F13	M	23.3947	−	149.9890	−	77.5576	−	**1.1362e−04**	+	0.0124
	SD	4.1958		0.0065		65.0442		**1.9047e−04**		0.0130
Total "−"			10		10		05		04	
Total "+"			00		00		00		05	
Total "≈"			03		03		08		04	
Dim=2000										
	B	00		2.0345e−136		**0**		2.9446e−110		**0**
F1	M	03	−	1.6579e−134	−	**0**	≈	7.8349e−104	−	**0**
	SD	1.0406e−84		3.1174e−134		**0**		1.7517e−103		**0**
	B	1.6118e−56		1.8041e−84		**0**		7.0031e−57		**0**
F2	M	3.3904e−52	−	1.0177e−57	−	**0**	≈	1.3902e−53	−	**0**
	SD	6.3223e−52		1.7220e−57		**0**		1.9649e−53		**0**
	B	2.6567e+08		2.8441e−134		**0**		1.3436e−73		**0**
F3	M	3.5043e+08	−	2.1675e−129	−	**0**	≈	8.1618e−45	−	**0**
	SD	8.6312e+07		4.8462e−129		**0**		1.8250e−45		**0**
	B	27.8055		3.5621e−69		**0**		4.0765e−57		**0**
F4	M	71.3256	−	2.7682e−67	−	**0**	≈	5.2805e−54	−	**0**
	SD	25.8691		2.3147e−67		**0**		5.6185e−54		**0**
	B	1.9833e+03		1.9989e+03		1.9787e+03		**0.0314**		1.0695
F5	M	1.9854e+03	−	1.9989e+03	−	1.9802e+03	−	**0.0801**	+	6.1011
	SD	1.6661		0.0105		2.2606		**0.0524**		6.5567
	B	44.8365		497.6830		10.9978		**5.4337e−05**		1.4007e−06
F6	M	67.6767	−	498.2376	−	27.3703	−	**4.7580e−04**	+	0.4242
	SD	17.2214		0.5606		17.8669		**3.2152e−04**		0.4444

(*continued on next page*)

Table 5 (*continued*).

Functions		WOA		TLBO		SGO		HHO		MSGO
		Results	WRS	Results	WRS	Results	WRS	Results	WRS	Results
F7	B	6.3908e−04		1.9017e−04		**5.1066e−05**		**4.3011e−06**		**7.3005e−06**
	M	0.0016	−	4.5662e−04	−	**7.2874e−05**	≈	**6.0770e−05**	≈	**5.6171e−05**
	SD	0.0012		2.4019e−04		**2.5063e−05**		**3.3666e−05**		**3.2942e−05**
F8	B	−8.3659e+05		−5.5400e+04		−3.9862e+05		**−8.3797e+05**		**−8.3797e+05**
	M	−8.1325e+05	−	−4.6507e+04	−	−2.8652e+05	−	**−8.3796e+05**	≈	**−8.3796e+05**
	SD	3.9476e+04		6.1120e+03		6.7616e+04		**2.8271**		**1.5266**
F9	B	**0**		**0**		**0**		**0**		**0**
	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
F10	B	**8.8818e−16**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**		**8.8818e−16**
	M	**2.3093e−15**	≈	**3.0198e−15**	≈	**8.8818e−16**	≈	**8.8818e−16**	≈	**8.8818e−16**
	SD	**1.9459e−15**		**1.9459e−15**		**0**		**0**		**0**
F11	B	**0**		**0**		**0**		**0**		**0**
	M	**0**	≈	**0**	≈	**0**	≈	**0**	≈	**0**
	SD	**0**		**0**		**0**		**0**		**0**
F12	B	0.0190		1.1717		0.0048		**1.6730e−08**		1.2209e−07
	M	0.0364	−	1.1742	−	0.0108	−	**2.2418e−07**	+	1.7067e−05
	SD	0.0177		0.0024		0.0053		**2.2682e−07**		2.2919e−05
F13	B	18.3687		199.9836		5.7699		**5.3914e−08**		0.0012
	M	23.3947	−	199.9894	−	10.2399	−	**5.1090e−05**	+	0.0239
	SD	4.1958		0.0041		5.9998		**4.0336e−05**		0.0410
Total "−"		B	10		10		05		04	
Total "+"			00		00		00		05	
Total "≈"			03		03		08		04	

" −", "+ " and "≈" denote that performance of algorithms are worse, better and similar to MSGO respectively.

for the function F22 in the place of best value, worse value, and median value, we have put star '*' because we think that these values are wrongly put in the paper [40]. For the function F28, the minimum function value is given as −1500 in the paper [40]. But we are getting less than that. So we are in confusion about the result. Thus we are excluding this result and have put '*' in place of test in Table 6 to avoid conflicts in future.

Finally, we conclude that MSGO proves to be very competitive against BOA in optimizing all 29 benchmark functions. In 10 functions, it is showing very promising results, and in rest 19 functions, it is fairing equally good with BOA. But one very positive note is MSGO converges faster than BOA since it is using very fewer numbers of FEs compared to BOA.

Experiment 3

In this experiment, SSOA (Squirrel Search Optimization Algorithm) [41] is compared with the MSGO algorithm. For comparison of the performances of both the algorithms, 26 benchmark functions are considered. These benchmark functions are described in the paper [41] and these benchmark functions are taken from [143,144]. Out of these 26 classical benchmark functions, four are unimodal separable, eight are unimodal nonseparable, six are multimodal separable and 8 are multimodal nonseparable benchmark functions. The results of SSOA are directly derived from [41] and we simulate results of MSGO with the same experimental set up like previous experiments.

The common control parameters such as pop_size are set as 50 and maximum iterations are set as 500 for the SSOA algorithm in its respective paper [41]. But for the MSGO algorithm, we have taken pop_size is 25, and the maximum iteration is 500. So max_FEs is set to 25,000(2 × 25×500 = 25,000, factor 2 is multiplied since fitness calculation is done two times for every iteration for each particle in population).

According to paper [41], for each function algorithms run 30 times with randomly generated populations. Similar to these experiments, the MSGO algorithm runs for 30 times with randomly generated populations. Statistical results in terms of mean value, corresponding standard-deviation, the best value, and worse value are reported in Tables 7–10, and the Results of the WRS test at a 0.05 significance level is conducted on the experimental results. The last three rows of each table summarize the results. For every benchmark function, the best results are boldfaced. Table 7 presents results for unimodal separable functions, Table 8 for unimodal nonseparable functions, Table 9 for multimodal separable and Table 10 for multimodal nonseparable functions.

Discussion

It is clear from Table 7 that MSGO achieves success in finding global optimum on unimodal separable functions F1, F2, and .F3. For F1, the performance of SSOA is found identical to MSGO. Neither MSGO nor SSOA finds the global optimum value for function F4. However, MSGO can achieve superior values compared to SSOA for this function. Table 8 provides results on unimodal nonseparable functions. It is clear from Table that MSGO achieves success in finding global optimum on functions F5, F6, F7, F9, F10, F11, but SSOA achieves success in finding only on F6. For F6 and F12, the performance of SSOA is found identical to MSGO. For the F8 SSOA outperforms MSGO. It is clear from Table 9 results that MSGO achieves success in finding global optimum on multimodal separable functions F13, F14, F15, F18, but SSOA achieves success in finding only on F13 and F15. For F13 and F15 only, the performance of SSOA is found identical to MSGO. Only for F16 and F17, MSGO could not reach the global optimum region but finding better results than the SSOA. It is clear from Table 10 results that MSGO achieves success in finding global optimum on multimodal nonseparable functions F19, F20, F21, F22, F23, F25, but SSOA achieves success in finding only on F20, F21, F22, and F23. For F13 and F15 only, the performance of SSOA is found identical to MSGO. Only for F24 and F26, MSGO could not reach the global optimum region but finding better mean results than the SSOA

From Table 7, we conclude that according to the WRS test, the MSGO algorithm is finding better results than the SSOA algorithm in 3 benchmark functions and equivalent result in one function out of 4 unimodal separable benchmark functions. Out of 8 unimodal nonseparable benchmark functions in 6 cases, the MSGO algorithm shows a better solution, in one case equivalent solution and one case worse solution in comparison to the SSOA

Table 6
Comparison on BOA and MSGO on 30 independent runs with 25,000 fitness function evaluations.

Function No	Function name	Algorithms	Mean	WRS results	Std	Best	Median	Worse
F1	Sphere	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F2	Beale	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F3	Cigar	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F4	Step	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F5	Quartic with noise	BOA	**3.8917e−05**	≈	**2.9003e−05**	**5.8800e−05**	**3.5850e−05**	**1.2000e−05**
		MSGO	**6.2169e−05**		**5.6360e−05**	**2.4126e−06**	**4.5817e−05**	**1.8305e−04**
F6	Bohachevsky	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F7	Ackley	BOA	1.7183e+00	−	0	1.7183e+00	1.7183e+00	1.7183e+00
		MSGO	**8.8818e−16**		**0**	8.8818e−16	8.8818e−16	8.8818e−16
F8	Griewank	BOA	1.8472e−19	−	2.6886e−20	1.6300e−19	1.6300e−19	2.1700e−19
		MSGO	**0**		**0**	**0**	**0**	**0**
F9	Levy	BOA	4.4108e−01	−	5.7467e−02	2.8900e−01	3.7800e−01	5.7400e−01
		MSGO	**4.6865e−04**		**3.9905e−04**	**3.6860e−05**	**3.5064e−04**	**1.3561e−03**
F10	Michalewicz	BOA	−5.3382e+00	−	−5.6092e+00	−5.7700e+00	−3.6200e+00	−2.2500e+00
		MSGO	−**8.2061**		0.6004	−9.4055	−8.1020	−7.2321
F11	Rastrigin	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F12	Alpine	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F13	Schaffer	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F14	Rosenbrock	BOA	2.8837e+01	−	3.1281e−02	2.8754e+01	2.8754e+01	2.8927e+01
		MSGO	**0.0068**		**0.0088**	**9.1662e−06**	**0.0036**	**0.0298**
F15	Easom	BOA	−**1**	≈	**0**	−**1**	−**1**	−**1**
		MSGO	−**1**		**0**	−**1**	−**1**	−**1**
F16	Shubert	BOA	−**1.8673e+02**	≈	**2.06493e−11**	−**1.8673e+02**	−**1.8673e+02**	−**1.8673e+02**
		MSGO	−**186.7309**		**4.7915e−14**	−**186.7309**	−**186.7309**	−**186.7309**
F17	Schwefel 1.2	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F18	Schwefel 2.21	BOA	6.9906e−153	−	1.4788e−152	3.7983e−155	2.3793e−153	9.7687e−152
		MSGO	**0**		**0**	**0**	**0**	**0**
F19	Schwefel 2.22	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F20	Schwefel 2.26	BOA	−2.2662e+03	−	4.5626e+02	−2.8790e+03	−2.3458e+03	−1.8373e+03
		MSGO	−**1.2569e+04**		**0**	−**1.2569e+04**	−**1.2569e+04**	−**1.2569e+04**
F21	Booth	BOA	**0**	≈	**0**	0	0	0
		MSGO	**0**		**0**	0	0	0
F22	Goldstein–Price	BOA	**3.0000**	≈	**0.0000e+00**	0.0000e+00*	0.0000e+00*	0.0000e+00*
		MSGO	**3.0000**		**0**	3.0000	3.0000	3.0000
F23	Matyas	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F24	Powell	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F25	Power sum	BOA	2.8400e−02	−	1.3179e−02	8.9600e−03	2.6800e−02	5.4300e−02
		MSGO	**2.2637e−04**		**5.4052e−03**	**1.2368e−09**	**6.3646e−04**	**0.0020**
F26	Shekel 4.5	BOA	−**1.0200e+01**	−	**0**	−**1.0200e+01**	−**1.0200e+01**	−**1.0200e+01**
		MSGO	−**10.1532**		**0**	−**10.1532**	−**10.1532**	−**10.1532**
F27	Sum squares	BOA	**0**	≈	**0**	**0**	**0**	**0**
		MSGO	**0**		**0**	**0**	**0**	**0**
F28	Trid	BOA	−2.7500e+07*	*	5.6500e+07*	−9.6100e+07*	−3.1800e+07*	7.3000e+07*
		MSGO	−1.4219e+03		121.9605	−1.5912e+03	−1.4319e+03	−1.0143e+03
F29	Zettl	BOA	−**3.7900e−03**	≈	**0**	−**3.7900e−03**	−**3.7900e−03**	−**3.7900e−03**
		MSGO	−**3.7900e−03**		**0**	−**3.7900e−03**	−**3.7900e−03**	−**3.7900e−03**
F30	Leon	BOA	1.1527e−06	−	9.4711e−07	1.0700e−08	6.9200e−07	2.6800e−06
		MSGO	**6.4571e−20**		**2.0169e−19**	**1.5407e−30**	**6.4581e−22**	**9.1054e−19**
Total "−"				10				
Total "+."				00				
Total "≈"				19				

" −", "+ " and "≈" denote that performance of GOA algorithms are worse, better and similar to MSGO respectively.

Table 7
Comparison on SSOA and MSGO on 30 runs on unimodal separable functions.

Function no.	Function name	Algorithms	Best	Worse	Mean	WRS results	Std
F1	Step	SSOA MSGO	**0** **0**	**0** **0**	**0** **0**	≈	**0** **0**
F2	Sphere	SSOA MSGO	7.9225e−20 **0**	5.7411e−07 **0**	4.1689e−08 **0**	−	1.4356e−07 **0**
F3	Sumsquares	SSOA MSGO	2.0052e−28 **0**	2.3194e−06 **0**	1.5201e−07 **0**	−	4.6741e−07 **0**
F4	Quartic	SSOA MSGO	3.0998e−02 **2.4126e−06**	9.9258e−01 **7.8155e−05**	5.0192e−01 **2.7811e−05**	−	2.9565e−01 **3.6713e−05**
Total "−"						03	
Total "+"						00	
Total "≈"						01	

" −", "+ " and "≈" denote that performance of SSOA algorithms are worse, better and similar to MSGO respectively.

Table 8
Comparison on SSOA and MSGO on 30 runs on unimodal nonseparable functions.

Function no.	Function Name	Algorithms	Best	Worse	Mean	WRS results	Std
F5	Beale	SSOA MSGO	2.1832e−29 **0**	2.7633e−20 **0**	9.5584e−22 **0**	−	5.0400e−21 **0**
F6	Easom	SSOA MSGO	−**1** −**1**	−**1** −**1**	**-1** **-1**	≈	**0** **0**
F7	Matyas	SSOA MSGO	1.5111e−29 **0**	2.0707e−24 **0**	1.542e−25 **0**	−	4.7571e−25 **0**
F8	Colville	SSOA MSGO	**8.5561e−21** 4.1891e−11	**2.4871e−08** 4.7812e−05	**1.4309e−09** 2.3012e−06	+	**4.6907e−09** 1.9012e−05
F9	Zakharov	SSOA MSGO	1.9954e−23 **0**	1.5225e−07 **0**	5.2215e−09 **0**	−	2.7772e−08 **0**
F10	Schwefel 2.22	SSOA MSGO	2.2266e−08 **0**	7.1423e−03 **0**	5.1849e−04 **0**	−	1.4144e−03 **0**
F11	Schwefel 1.2	SSOA MSGO	6.6803e−18 **0**	3.4907e−04 **0**	1.6925e−05 **0**	−	6.6811e−05 **0**
F12	Dixon–Price	SSOA MSGO	**1.8308e−01** **0.1256**	**6.6951e−01** **0.2627**	**2.2412e−01** **0.2137**	−	**1.2107e−01** **0.0412**
Total "−"						06	
Total "+"						01	
Total "≈"						01	

" −", "+ " and "≈" denote that performance of SSOA algorithms are worse, better and similar to MSGO respectively.

Table 9
Comparison on SSOA and MSGO on 30 runs on multimodal separable functions.

Function No.	Function Name	Algorithms	Best	Worse	Mean	WRS results	Std
F13	Bohachevsky1	SSOA MSGO	**0** **0**	**0** **0**	**0** **0**	≈	**0** **0**
F14	Booth	SSOA MSGO	1.2622e−29 **0**	8.1255e−24 **0**	9.5859e−25 **0**	−	1.7996e−24 **0**
F15	Michalewicz2	SSOA MSGO	−**1.8013** −**1.8013**	−**1.8013** −**1.8013**	−**1.8013** −**1.8013**	≈	**1.0275e−15** **2.1446e−16**
F16	Michalewicz5	SSOA MSGO	−**4.6877** −**4.6877**	−3.5563 −4.5249	−4.3479 −**4.5284**	−	3.2785e−01 **0.0432**
F17	Michalewicz10	SSOA MSGO	−9.4806 −9.5515	−5.9146 −7.0138	−7.5900 −**8.2522**	−	9.9570e−01 0.6163
F18	Rastrigin	SSOA MSGO	**0** **0**	7.6657e−06 **0**	4.9059e−07 **0**	−	1.5057e−06 **0**
Total "−"						04	
Total "+"						00	
Total "≈"						02	

" −", "+ " and "≈" denote that performance of SSOA algorithms are worse, better and similar to MSGO respectively.

algorithm. Out of 6 multimodal separable benchmark functions in 4 cases, the MSGO algorithm shows a better solution, in two cases equivalent solution with SSOA algorithm. Out of 8 multimodal nonseparable benchmark functions in 4 cases MSGO algorithm shows a better solution in a 4 case equivalent solution with an SSOA algorithm.

So overall, only in 1 case out of 26 functions, MSGO performs worse than the SSOA algorithm. All in other cases, MSGO

Table 10
Comparison on SSOA and MSGO on 30 runs on multimodal nonseparable functions.

Function no.	Function name	Algorithms	Best	Worse	Mean	WRS results	Std
F19	Schaffer	SSOA MSGO	0 **0**	9.7159e−03 **0**	9.7159e−04 **0**	−	2.9646e−03 **0**
F20	Six Hump Camel Back	SSOA MSGO	−**1.03163** −**1.03163**	−**1.03163** −**1.03163**	−**1.03163** −**1.03163**	≈	**4.5168e−16** **0**
F21	Boachevsky2	SSOA MSGO	**0** **0**	**0** **0**	**0** **0**	≈	**0** **0**
F22	Boachevsky3	SSOA MSGO	**0** **0**	**0** **0**	**0** **0**	≈	**0** **0**
F23	Shubert	SSOA MSGO	−**186.73** −**186.73**	−**186.73** −**186.73**	−**186.73** −**186.73**	≈	**2.6389e−14** **1.6722e−15**
F24	Rosenbrock	SSOA MSGO	3.9637e−19 4.7812e−06	2.8475e+01 0.0138	9.4919e−01 **0.0068**	−	5.1988e+00 **0.0088**
F25	Griewank	SSOA MSGO	**0** **0**	4.1375e−05 **0**	3.435e−06 **0**	−	9.6702e−06 **0**
F26	Ackley	SSOA MSGO	2.2418e−10 **8.8818e−16**	2.5867e−03 **8.8818e−16**	1.3915e−04 **8.8818e−16**	−	4.8513e−04 **0**
Total "−" Total "+" Total "≈"						04 00 04	

" −", "+ " and "≈" denote that performance of SSOA algorithms are worse, better and similar to MSGO respectively.

performs either better or equivalent solution in comparison with the SSOA algorithm. So we conclude that the MSGO algorithm has outperformed the SSOA algorithm.

Experiment 4

In this experiment, the VPL (Volleyball Premier League) algorithm [70] is compared with the MSGO algorithm. For comparison of the performance of algorithms, 23 classical benchmark functions are considered. These benchmark functions are described in Appendix A. Here, VPL algorithm results are reported from paper [70].

The common control parameter, such as max_FEs is set to 100,000 for the VPL algorithm in its own paper [70]. Hence for the MSGO algorithm, the pop_size is taken as 50, and the maximum iteration is 1000. So max_FEs is $2 \times 50 \times 1000 = 100{,}000$, factor 2 is multiplied since the fitness calculation is done two times for every iteration for each particle in population.

According to paper [70], for each function, algorithms run 30 times with randomly generated populations. Similar to that experiment, here MSGO algorithm runs for 30 times with randomly generated populations. Statistical results in terms of mean value and corresponding standard deviation, the best value, and worse value are reported in Tables. Results are reported in Table 11 for unimodal, Table 12 for multimodal, Table 13 for multimodal fixed dimensional multimodal benchmark functions. For every benchmark function, the best results are boldfaced. A result of the WRS test at a 0.05 significance level is conducted on experimental results, and the last three rows summarize results.

Discussion

According to Table 11, MSGO has gained the best performance and consistently performed better than VPL algorithms. MSGO has an excellent performance in exploitation and convergence then VPL, and it successfully overcomes to solve all the problems within this category. It is clear from the results that MSGO achieves success in finding global optimum on F1–F4. For F5, F6, and F7, the performance of MSGO is better than VPL. Form Table 12 results, we find that the MSGO algorithm is eligible for solving problems with challenging search space. It is clear from results that MSGO achieves success in finding global optimum on F8, F9, F11, F12, and F13, whereas VPL achieves success for F9 and F11. For F10, both MSGO and VPL get equivalent results. The results reported from Table 13 for functions F14–F23 indicate that MSGO has an excellent exploration capability. The Table shows that MSGO has consistently performed better than VPL algorithms. It is clear from table results that MSGO algorithm achieves success in finding global optimum on F14, F16, F17, F18, F19, F20, F21, F22, F23.and for F15, almost reach to global optimum (best value reach to global optimum, but mean value reach near-global optimum). In contrast, VPL reaches optimal solution only for F14, F16, F17, and F18.

From Table 11, we see that in all the cases of unimodal benchmark functions, the MSGO algorithm shows better results than the VPL algorithm. Similarly comparison on best results obtained by algorithms, we get that MSGO algorithm either gets the best result than the VPL algorithm or similar result with the VLP algorithm, So we can say that the MSGO algorithm is showing the best performance in solving unimodal benchmark functions in compare to VPL algorithms. In Table 12, there is '*' mark in the first row with the result of the VPL algorithm to say that there result may be put wrongly because the optimal value of the F8 function is −12569.487. The MSGO algorithm can find the optimum solution for F8. So for comparison, we have considered only five multimodal functions. According to the WRS test from Table 12, we get that in two cases out of five cases. The MSGO algorithm shows better performance than the VPL algorithm and similar to 3 cases out of five cases with the VPL algorithm. Similarly comparison on best results obtained by algorithms, we get that the MSGO algorithm either gets the best result than the VPL algorithm or similar result with the VPL algorithm. So we can say that the MSGO algorithm is showing the best performance in solving multimodal benchmark functions in comparison to VPL algorithms. In Table 13, there is '*' mark with the result of the VPL algorithm to say that there result might be put wrongly as the optimum value for the F15 function is 3.0749E−04. According to WRS test from Table 13, we get that in six cases out of 10 cases MSGO algorithm shows better performance than VPL algorithm and similar with 4 cases out of 10 cases with VPL algorithm Similarly comparison on best results obtained by algorithms we get that MSGO algorithm either get the best result than VPL algorithm or similar result with VLP algorithm. So we can say that the MSGO algorithm is showing the best performance in solving fixed dimensional multimodal benchmark functions in comparison to VPL algorithms.

Table 11
Result of unimodal benchmark functions.

Function no.	Algorithms	Best	Worse	Mean	Std	WRS result
F1	VPL	0	2.34e−130	7.81e−132	4.20e−131	−
	MSGO	**0**	**0**	**0**	**0**	
F2	VPL	1.12e−102	2.85e−89	1.13e−90	5.13e−90	−
	MSGO	**0**	**0**	**0**	**0**	
F3	VPL	1.93e−33	1.53e−02	8.16e−04	2.85e−03	−
	MSGO	**0**	**0**	**0**	**0**	
F4	VPL	0.00e+00	1.63e−28	1.54e−29	3.96e−29	−
	MSGO	**0**	**0**	**0**	**0**	
F5	VPL	2.58e+01	2.67e+01	2.62e+01	2.76e−01	−
	MSGO	**4.5282e−06**	**0.0163**	**0.0030**	**0.0048**	
F6	VPL	1.82e−05	2.34e−03	4.09e−04	5.33e−04	−
	MSGO	**5.2869e−07**	**5.2869e−05**	**2.2673e−06**	**2.3899e−05**	
F7	VPL	4.67e−05	4.81e−03	1.93e−03	1.36e−03	−
	MSGO	**5.9024e−07**	**3.6213e−05**	**1.0034e−05**	**9.5686e−06**	
Total +						00
Total −						00
Total ≈						07

“ −”, “+ ” and “≈” denote that performance of VPL is worse, better and similar to MSGO respectively.

Table 12
Result of Multimodal Benchmark Functions.

Function No.	Algorithms	Best	Worse	Mean	Std	WRS results
F8	VPL	−1.19e+112*	−7.38e+90*	−4.68e+110*	2.15e+111*	----
	MSGO	**−1.2569e+04**	**−1.2569e+04**	**−1.2569e+04**	**5.9179e−04**	
F9	VPL	**0**	**0**	**0**	**0**	≈
	MSGO	**0**	**0**	**0**	**0**	
F10	VPL	**8.88e−16**	**8.88e−16**	**8.88e−16**	**9.86e−32**	≈
	MSGO	**8.8818e−16**	**8.8818e−16**	**8.8818e−16**	**0**	
F11	VPL	**0**	**0**	**0**	**0**	≈
	MSGO	**0**	**0**	**0**	**0**	
F12	VPL	1.11e−06	6.56e−05	2.58e−05	1.74e−05	−
	MSGO	**3.4833e−10**	**1.3494e−06**	**1.3505e−07**	**2.9843e−07**	
F13	VPL	2.63e−05	2.31e−03	4.18e−04	4.84e−04	−
	MSGO	**3.4528e−09**	**1.4118e−04**	**1.4357e−05**	**2.8625e−05**	
Total −						02
Total +						00
Total ≈						03

“ −”, “+ ” and “≈” denote that performance of VPL is worse, better and similar to MSGO respectively and.

Experiment 5

In this experiment, SELO (Socio-Evolution and Learning-Optimization) [71] algorithm is compared with the MSGO algorithm. For comparison of the performance of algorithms, 50 benchmark functions are considered. These sets of test functions include unimodal, multimodal, separable, and non-separable [145–147]. All benchmark test problems are divided into four categories, such as US, MS, UN, and MN, and its range, formulation, characteristics, and dimensions of these problems are listed in paper [71]. Results of SELO is directly fetched from [71], but we simulate MSGO.

The common control parameter, such as maximum iterations is set as 70,000 for SELO algorithm in its respective paper [71]. But for the MSGO algorithm, we have taken pop_size is 25, and the maximum iteration is 500. So max_FEs is set to 25,000(2 × 25×500=25,000, factor 2 is multiplied since fitness calculation is done two times for every iteration for each particle in population.

According to paper [71], for each function, algorithms run 30 times with randomly generated populations. In this experiment, also we have run our simulations for 30 runs with randomized populations. Statistical results in terms of mean value and corresponding standard deviation and best value are reported in Table 14. In this experiment values below $1E^{-16}$ are considered to be zero. For every benchmark function, the best results are boldfaced. To have statistically sound conclusions, the WRS test at 0.05 significance level was conducted on experimental results, and the last three rows of the table summarize results.

Discussion

From Table 14, it is observed that the MSGO algorithm achieves success in finding global optimum on F1, F5, F11, F14–F18, F25, F30, F33, F35–F47, F50 functions and near-global optimum for functions F2, F19, F20, F21, F26. But for function F3, F4, F6, F12–F14, F19, F21, F22, F27–F29, F36, F37, F48, F49 neither MSGO nor SELO find optimal results. However, MSGO can derive better results than SELO for these functions. SELO can outperform MSGO in F23, F24, F31, F34 functions in finding better results.

In Table 14, there is ‘*’ against the result of the SELO algorithm for the function F14, and F26, since we guess that the result might have been mentioned wrongly as the minimum value of F14 function, is −1, and minimum value for F26 is −1.801303410098554. We have considered F14 for comparison purposes since the best value is defined, but we have omitted the comparison for F26 in our table. Hence, a total of 49 functions are taken into consideration for the analysis of performances. Overall, we conclude that the MSGO algorithm shows better or competitive performance than the SELO algorithm. However, the convergence of MSGO is faster than SELO since it takes much fewer FEs than SELO (may

Table 13
Results on fixed-dimensional benchmark functions.

Function no.	Algorithms	Best	Worse	Mean	Std	WRS results
F14	VPL MSGO	**9.98e−01** **0.9980**	**9.98e−01** **0.9980**	**9.98e−01** **0.9980**	**2.32e−13** **0**	≈
F15	VPL MSGO	2.45e−05* **3.0749e−04**	1.82e−03 **4.2429e−04**	1.25e−03 **3.2093e−04**	3.08e−04 **3.2667e−05**	−
F16	VPL MSGO	**−1.03e+00** **−1.0316**	**−1.03e+00** **−1.0316**	**−1.03e+00** **−1.0316**	**2.56e−06** **0**	≈
F17	VPL MSGO	**3.98e−01** **0.3979**	**3.98e−01** **0.3979**	**3.98e−01** **0.3979**	**2.69e−06** **0**	≈
F18	VPL MSGO	**3.0000** **3.0000**	**3.0000** **3.0000**	**3.0000** **3.0000**	**7.58e−05** **0**	≈
F19	VPL MSGO	−3.85e+00 **−3.8628**	−3.49e+00 **−3.8628**	−3.77e+00 **−3.8628**	9.37e−02 **0**	−
F20	VPL MSGO	−3.32e+00 **−3.3220**	−3.20e+00 **−3.3220**	−3.28e+00 **−3.3220**	5.41e−02 **2.1453e−16**	−
F21	VPL MSGO	−1.02e+01 **−10.1532**	−5.06e+00 **−10.1532**	−9.30e+00 **−10.1532**	1.90e+00 **0**	−
F22	VPL MSGO	−1.04e+01 **−10.4029**	−5.09e+00 **−10.4029**	−8.99e+00 **−10.4029**	2.35e+00 **0**	−
F23	VPL MSGO	−1.05e+01 **−10.5364**	−3.57e+00 **−10.5364**	−9.40e+00 **−10.5364**	2.28e+00 **0**	−
Total −						06
Total +						00
Total ≈						04

" −", "+ " and "≈" denote that performance of VPL is worse, better and similar to MSGO respectively.

be noted here that max_FEs for MSGO is 25,000 and maximum iteration for SELO is 70,000).

Experiment 6

In this experiment, the GROM [92] algorithm is compared with the MSGO algorithm. For comparison of the performance of algorithms, 23 classical benchmark functions are considered. These benchmark functions are described in Appendix A. Here, test functions are solved in two cases, including low dimensional and high dimensional. For results MSGO algorithm, the codes are implemented by us. For the GROM algorithm, the results are taken from [92].

For low dimensional case, the common control parameters such as population size are set to 40, maximum iterations 500, and maximum number function evaluations are set to 40,000. For each function, algorithms run 30 times with different randomly generated populations. Statistical results in terms of best value, mean value, and corresponding standard deviation are reported in Tables 15–17. Table 15 reports results for Unimodal benchmark functions, Table 16 for Multimodal benchmark functions, Table 17 for Fixed dimensional multimodal benchmark functions.

For high dimensional case, the 200-dimensional version of unimodal and multimodal test functions are solved in two cases. In case 1, pop_size is set as 200, and the maximum iteration is 2000, and the results are given in Table 18. In case 2, pop_size is taken as 40, and the maximum iteration is 500, and the results are given in Table 19.

Here too, to have statistically sound conclusions, the WRS test at a 0.05 significance level was conducted on experimental results, and the last three rows of each respective table summarize experimental results.

Discussion

Simulations on MSGO, from Table 15, It is clear that MSGO achieves success in finding global optimum on F1–F4. For F5 and F7, the performance of MSGO is better than GROM, and for F6, GROM is better than the MSGO algorithm. Form Table 16 MSGO achieves success in finding global optimum on F8, F9, and F11, whereas GROM achieves success for F9 and F11. For F10, both MSGO and GROM get equivalent results. For F8 and F13, the MSGO algorithm finds better results than GROM, and GROM finds better than MSGO for F12 function. The results reported from Table 17 for functions F14–F23 indicate that both MSGO and GROM have an excellent exploration capability. It is clear from table results that both GROM and MSGO algorithm achieve success in finding global optimum on all functions except F14. For F14, MSGO achieves success, whereas GROM is not in finding a globally optimal solution. From Tables 18 and 19, it is clear that for both the cases MSGO algorithm finds a superior solution than GROM for all 13 functions except F9, F10, and F11. For F1, F2, F3, F4, F9, and F11, the MSGO algorithm achieves success in finding a global optimal solution whereas only for F9 and F11, GROM achieves success.

From Table 15, we conclude that out of seven unimodal test functions, the MSGO performs better than GROM in six test functions and worse in one test function. From Table 16, we summarize that out of six multimodal tests, MSGO performs better than the GROM in two functions, equivalent in three test functions and worse in one test function. And from Table 17, out of 10 fixed dimensional multimodal test functions, the MSGO performs better than the GROM in one test function and equivalent with nine test functions. For high dimensional cases, that is from Table 18, in the case of a 200-dimensional version of unimodal and multimodal test functions with population size 200, maximum iteration 2000, the MSGO performs better than GROM in 10 test functions, equivalent with three test functions out 13 test functions. Similarly, from Table 19 in the case of a 200-dimensional version of unimodal and multimodal test functions with pop_size 40, maximum iteration 500, the MSGO performs better than the GROM in 10 test functions and equivalent with three tests functions out of 13 test functions. Considering the above findings, we can summarize that MSGO does much better compared to GROM.

Experiment 7

After having simulated, compared, and analyzed the performance of MSGO against many old as well as newer meta-heuristic algorithms and proving the superiority of MSGO over others, in

Table 14
Comparison on SELO and MSGO on 30 repetitions.

Function no.	Function name	Algorithms	Mean	WRS results	Std	Best
F1	Foxholes	SELO	**0.9980038538690870**	≈	**0.0000013769725269**	**0.9980038383022720**
		MSGO	**0.998003837794450**		**0**	**0.998003837794450**
F2	Goldstein–Price	SELO	**3.0013971187248700**	≈	**0.0018936009191261**	**3.0000021202023800**
		MSGO	**2.999999999999924**		**1.704061088757890e−15**	**2.999999999999921**
F3	Penalized	SELO	0.2899597890213580	−	0.0159272187796787	0.2497905224307240
		MSGO	**2.535263418316779e−05**		**3.588526033613978e−05**	**3.691147653138041e−09**
F4	Penalizd2	SELO	2.3720510573781100	−	0.1531241868389090	2.0664368584658500
		MSGO	**3.917682943046329e−04**		**7.076007053872754e−04**	**1.486170002143908e−08**
F5	Ackley	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0**		**0**	**0**
F6	Beale	SELO	0.0000997928359263	−	0.0001311815541321	0.0000007530509495
		MSGO	**0**		**0**	**0**
F7	Bonachevsky1	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0.0000000000000000**		**0.0000000000000000**	**0.0000000000000000**
F8	Bonachevsky2	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0.0000000000000000**		**0.0000000000000000**	**0.0000000000000000**
F9	Bonachevsky3	SELO	**0.0000000000000000**	≈	**0.0000000000000001**	**0.0000000000000000**
		MSGO	**0.0000000000000000**		**0.0000000000000000**	**0.0000000000000000**
F10	Booth	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0.0000000000000000**		**0.0000000000000000**	**0.0000000000000000**
F11	Branin	SELO	**0.3978943993817670**	≈	**0.0003536060523484**	**0.3978822494361650**
		MSGO	**0.397887357729738**		**0**	**0.397887357729738**
F12	Colville	SELO	3.6688019971758100	−	1.7577708967227600	0.8388908577815620
		MSGO	**9.056293802405678e−05**		**1.968055752300307e−04**	**1.975105144423035e−09**
F13	Dixon–Price	SELO	0.9737369841168760	−	0.0054869670667257	0.9541730938494050
		MSGO	**0.206088179706563**		**0.047550281561434**	**0.092227038909637**
F14	Easom	SELO	0.0000000000000000*	−	0.1083854312160620	0.5936514558196160
		MSGO	**−1**		**0**	**−1**
F15	Fletcher	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0.0000000000000000**		**0.0000000000000000**	**0.0000000000000000**
F16	Fletcher	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0.0000000000000000**		**0.0000000000000000**	**0.0000000000000000**
F17	Fletcher	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0.0000000000000000**		**0.0000000000000000**	**0.0000000000000000**
F18	Griewank	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0**		**0**	**0**
F19	Hartman3	SELO	−2.2922815000937700	−	0.5795350381767260	−3.5841184056629400
		MSGO	**−3.862782147820753**		**2.682341420777607e−15**	**−3.862782147820756**
F20	Hartman6	SELO	**−3.322301466912912**	≈	**0.0003690446342091**	**−3.322368011415481**
		MSGO	**−3.322368011415514**		**0.000097660317011**	**−3.322368011415516**
F21	Kowalik	SELO	0.0003493601571991	−	0.0000226057336871	0.0003226283751593
		MSGO	**3.0742855341950336e−04**		**7.881617660825124e−05**	**3.075006625965705e−04**
F22	Langermann2	SELO	−1.0835400071766800	−	0.5277882902242550	−2.1933014645645000
		MSGO	**−1.805899685010939**		**0.001916427981154**	**−1.816046503083688**
F23	Langermann5	SELO	**−1.4999998390866700**	+	**0.0000000818646069**	**−1.4999999590992100**
		MSGO	−0.681561202156719		0.222457661970647	−1.096697862060463
F24	Langermann10	SELO	**−1.4999991427332700**	+	**0.0000003717669841**	**−1.4999999303979900**
		MSGO	−0.318557792223066		0.275845685945768	−1.096671262013812
F25	Matyas	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0**		**0**	**0**
F26	Michalewicz2	SELO	−1.8166465888521900*	*	0.0072804985619476	−1.8106292157333700*
		MSGO	−1.801303410098555		9.033620566533923e−16	−1.801303410098554
F27	Michalewicz5	SELO	−3.3591408962129900	−	0.2009584117455920	−3.9631157953194900
		MSGO	**−4.434809535764638**		**0.183994472695330**	**−4.687657110410353**
F28	Michalewicz10	SELO	−3.9793838974626000	−	0.0005104314209355	−3.9806353395021300
		MSGO	**−8.230671230170975**		**0.699403314103083**	**−9.353040564526447**
F29	Perm	SELO	2.0169277899221400	−	1.2374893392409200	0.3208703882956150
		MSGO	**0.370522595914268**		**0.631667871073207**	**3.974325738298966e−04**
F30	Powell	SELO	**0.0000000000000000**	≈	**0.0000000000000000**	**0.0000000000000000**
		MSGO	**0**		**0**	**0**

(*continued on next page*)

Table 14 (*continued*).

Function no.	Function name	Algorithms	Mean	WRSresults	Std	Best
F31	Powersum	SELO MSGO	**0.0000000000000000** **0.004753019900010**	+	**0.0000000000000000** **0.009342177329505**	**0.0000000000000000** **2.244454547213270e−08**
F32	Quartic	SELO MSGO	**0.0000989055208389** **4.027999192659648e−05**	≈	**0.0000521772789680** **3.974861484903702e−05**	**0.0000104209894311** **8.780098361357959e−07**
F33	Rastrigin	SELO MSGO	**0.0000000000000000** **0**	≈	**0.0000000000000000** **0**	**0.0000000000000000** **0**
F34	Rosenbrock	SELO MSGO	**0.0000000000000000** 0.016277990982918	+	**0.0000000000000000** 0.024337997169173	**0.0000000000000000** 2.607105426422528e−06
F35	Schaffer	SELO MSGO	**0.0000000000000000** **0**	≈	**0.0000000000000000** **0**	**0.0000000000000000** **0**
F36	Schwefel	SELO MSGO	−0.3402784042291390 −**1.2569e+04**	−	3.2212919091274600 **0**	−0.0325083488969540 −**1.2569e+04**
F37	Schwefel 1.2	SELO MSGO	0.0000000000000009 **0**	−	0.0000000000000001 **0**	0.0000000000000007 **0**
F38	Schwefel 2.22	SELO MSGO	**0.000000000000000** **0**	≈	**0.000000000000000** **0**	**0.000000000000000** **0**
F39	Shekel10	SELO MSGO	−**10.536281667618100** −**10.536409816692050**	≈	**0.0000481237097736** **1.776356839400251e−15**	−**10.5363928369535000** −**10.536409816692050**
F40	Shekel5	SELO MSGO	−**10.1531669871808000** −**10.153199679058224**	≈	**0.0000172333322304** **7.067040111004328e−15**	−**10.1531973132210000** −**10.153199679058231**
F41	Shekel17	SELO MSGO	−**10.4028748144797000** −**10.402940566818664**	≈	**0.0000478046191696** **1.581958664619025e−15**	−**10.4029869270437000** −**10.402940566818666**
F42	Shubert	SELO MSGO	−**186.7153981691330000** −**1.867309088310239e+02**	≈	**0.0190762312882078** **2.350959084277836e−14**	−**186.7363874875390000** −**1.867309088310239e+02**
F43	Six ump camel back	SELO MSGO	−**1.0303924506027700** −**1.031628453489878**	≈	**0.0025133845110030** **1.440815654614292e−16**	−**1.0314918740874000** −**1.031628453489878**
F44	Sphere2	SELO MSGO	**0.000000000000000** **0**	≈	**0.000000000000000** **0**	**0.000000000000000** **0**
F45	Step2	SELO MSGO	**0.0000000000000010** **0**	≈	**0.0000000000000001** **0**	**0.0000000000000006** **0**
F46	Stepint	SELO MSGO	**0.0000000000000000** **0**	≈	**0.0000000000000000** **0**	**0.0000000000000000** **0**
F47	Sumsquares	SELO MSGO	**0.0000000000000000** **0**	≈	**0.0000000000000000** **0**	**0.0000000000000000** **0**
F48	Trid6	SELO MSGO	−46.672022811734100 −**47.484581750100304**	−	1.1721159591339900 **1.693041691527924**	−48.933141750726300 −**49.613835674928339**
F49	Trid10	SELO MSGO	−162.571266865506000 −**2.099999999933476e+02**	−	0.2649613601835890 **1.252363639681460e−08**	−162.922114827822000 −**2.100000000000009e+02**
F50	Zakharov	SELO MSGO	**0.0000000000000000** **0**	≈	**0.0000000000000000** **0**	**0.0000000000000000** **0**
Total +				04		
Total −				16		
Total ≈				29		

" −−", "+ " and "≈" denote that performance of SELO is worse, better and similar to MSGO respectively.

this experiment, we have applied this to solve a real-world problem. This experiment is aimed at solving short-term hydrothermal scheduling problems. The objective of short-term scheduling of a hydrothermal power system is to schedule power generations of thermal and hydro units in system to satiate load demands, in scheduling duration of 24 h, in combination with various constraints on hydraulic systems and power system networks. The objective function of the hydrothermal scheduling problem is to be minimized the overall fuel cost of thermal units for a given short term. The hydrothermal system considered here is extremely complex and involves nonlinear relationships of decision variables, cascaded nature of the hydraulic network, water carry delays, and time link between consecutive schedules. Detail of the problem is given in [4]. This system consists of a multi-chain cascade of four hydro units and thermal units represented by an equivalent thermal plant and scheduling period of 24 h. The studied problem may be classified into three categories depending on the type of their fuel cost functions and constraints. Category 1 represents HTS problems with quadratic cost functions, Category 2 represents HTS problem with quadratic cost functions and prohibited discharge zone and Category 3 represents HTS problem considering the valve point effect and prohibited discharge zone. We have applied MSGO to all these categories and compared them with other algorithms that have already been used to solve the problem in respective literature as given in proceedings theories. In all our experiments, we have kept the maximum number of function evaluations as 150,000.

Case 1 (HTS problems with quadratic cost functions) In this case, the cost functions of the thermal units of the hydrothermal system are considered to be quadratic and there is no prohibited discharge zone of the reservoirs of the hydro plants. The result obtained by MSGO algorithm is compared with the reported results by other methods, namely GA With Multi-Parent Crossover (GA-MPC) [134], hybridization of a competitive variant of DE and a Local Search method (CDELS) [135], DE with the explorative ability of a Random Hill Climber(DE–RHC) [136], Real-Coded, Elite preserving GA (RCEP-GA) [137], Hybrid EA–DE Memetic Algorithm (EA-DE-MA) [138], Self Adaptive Cluster-based and Weed Inspired DE algorithm (SACWIDE) [139], Simulated Binary Crossover (SBX) with adaptive mutation (MSBX_AM) [140],

Table 15
Result of unimodal benchmark functions on repetitions.

Function no.	Algorithms	Best	Mean	Std	WRS results
F1	GROM MSGO	**0** **0**	3.1350e- 247 **0**	6.5961e−251 **0**	−
F2	GROM MSGO	2.8836e−53 **0**	1.367e−52 **0**	9.173e−53 **0**	−
F3	GROM MSGO	**0** **0**	3.398e−87 **0**	9.6154e−87 **0**	−
F4	GROM MSGO	7.155e−145 **0**	1.118e−131 **0**	3.0352e−132 **0**	−
F5	GROM MSGO	18.46243 **1.0879e−04**	19.12 **0.0467**	0.6160 **0.0568**	−
F6	GROM MSGO	**1.594e- 7** 2.0310e−07	**3.9096e- 7** 2.0310e−4	**3.573e- 7** 0.0012	+
F7	GROM MSGO	2.677e−5 **2.3567e−06**	0.000131 **2.8174e−05**	0.000131 **1.7922e−05**	−
Total + Total − Total ≈					01 06 00

" −", "+ " and "≈" denote that performance is worse, better and similar to MSGO, respectively.

Table 16
Result of multimodal benchmark functions.

Function no.	Algorithms	Best	Mean	Std	WRS results
F8	GROM MSGO	−9467.296 **−1.2569e+04**	−9070.8 **−1.2569e+04**	347.215 **0.0061**	−
F9	GROM MSGO	**0** **0**	**0** **0**	**0** **0**	≈
F10	GROM MSGO	**8.8817e−16** **8.8817e−16**	**8.8817e−16** **8.8817e−16**	**0** **0**	≈
F11	GROM MSGO	**0** **0**	**0** **0**	**0** **0**	≈
F12	GROM MSGO	**3.8742e−9** 2.4170e−08	**9.980e−9** 8.3541e−06	**5.7890e−9** 1.5855e−05	+
F13	GROM MSGO	**5.2761e−7** 7.0156e−09	**0.003506** 9.5875e−05	**0.006430** 2.4259e−04	−
Total + Total − Total ≈					01 02 03

" −", "+ " and "≈" denote that performance is worse, better and similar to MSGO respectively.

Cuckoo Search (CS) [141], Artificial Bee Colony (ABC) [141], DE algorithm with ensemble of parameters and mutation strategies (EPSDE) [141]. All results for this case 1 are given in Table 20 and the best result among all is highlighted in boldface. In table "NC" represents that particular result is not calculated in that experiment. The cost convergence characteristic of MSGO is given in Fig. 3.

From Table 20, it can be observed that MSGO gives an improved reduction in cost compared to other algorithms. The convergence characteristic of MSGO is smooth and settles for optimal value at 250 iterations, as shown in Fig. 3.

Case 2: **(HTS problem with quadratic cost functions and prohibited discharge zone)**

In this case, the cost functions of thermal units of the hydrothermal system are considered to be quadratic and there is a prohibited discharge zone of the reservoirs of the hydro plants. The result obtained by the MSGO algorithm is compared with the reported results by other methods, as stated in Case 1. Results are tabulated in Table 21 with the best result made bold and cost convergence, also shown in Fig. 4. In table "NC" represents that particular result is not calculated in that experiment.

In this case, MSGO does not perform as expected. GA-MPC gives the best result. MSGO is not able to converge smoothly, as shown in Fig. 4.

Case 3: **(HTS problem considering valve point effect and prohibited discharge zone)**

To check the feasibility of the proposed method for an effective system, in addition to prohibited discharge constraints, the valve point effect of the thermal generator is taken into consideration in this case, and results are tabulated in Table 22 with a convergence graph of MSGO in Fig. 5. In table "NC" represents that particular result is not calculated in that experiment.

Analyzing Table 22 and Fig. 5, it can be summarized that MSGO can give a better reduction in cost, and its convergence is also smooth. This experiment proves that MSGO is an effective metaheuristic algorithm not only to optimize benchmark functions but also to deal with the physical problem at hand.

6. Conclusion

Most real-world optimization problems are highly nonlinear, multimodal and multi-dimensional having various complex constraints. They might have different objectives, and sometimes they are conflicting with each other. Even it has been

Table 17
Results on fixed-dimensional benchmark functions.

Function no.	Algorithms	Best	Mean	Std	WRS results
F14	GROM MSGO	0.998 **0.9980**	1.1964 **0.9980**	0.5952 **0**	–
F15	GROM MSGO	**0.0003074** **3.0749e−04**	**0.0003074** **3.0749e−04**	**2.83762E−19** **1.7262e−15**	≈
F16	GROM MSGO	**−1.0316** **−1.0316**	**−1.0316** **−1.0316**	**3.315148E−7** **6.7752e−16**	≈
F17	GROM MSGO	**0.397887** **0.3979**	**0.397887** **0.3979**	**6.678E−7** **0**	≈
F18	GROM MSGO	**2.999** **3.0000**	**2.999** **3.0000**	**7.02166E−16** **1.0971e−15**	≈
F19	GROM MSGO	**−3.8627** **−3.8628**	**−3.862** **−3.8628**	**8.88178E−1** **2.1233e−16**	≈
F20	GROM MSGO	**−3.3219** **−3.3220**	**−3.29821** **−3.3220**	**0.0475558** **2.9012e−15**	≈
F21	GROM MSGO	**−10.153** **−10.1532**	**−10.1531** **−10.1532**	**7.944109E−16** **6.9035e−15**	≈
F22	GROM MSGO	**−10.402** **−10.4029**	**−10.402** **−10.4029**	**1.4862E−15** **8.0799e−16**	≈
F23	GROM MSGO	**−10.536** **−10.5364**	**−10.53** **−10.5364**	**1.776E−15** **2.4240e−15**	≈
Total +					00
Total −					01
Total ≈					09

" −", "+ " and "≈" denote that performance is worse, better and similar to MSGO respectively.

Table 18
Results on 200 dimensional Benchmark functions with pop_size 200 with 2000 iteration.

Function no.	Algorithms	Best	Mean	Std	WRS results
F1	GROM MSGO	4.0197e−283 **0**	5.0162e−283 **0**	0 **0**	–
F2	GROM MSGO	1.1623e−156 **0**	1.6258e−156 **0**	3.2777e−157 **0**	–
F3	GROM MSGO	1.2908e−207 **0**	1.7886e−191 **0**	0 **0**	–
F4	GROM MSGO	6.8726e−122 **0**	7.808e−122 **0**	7.3555e−123 **0**	–
F5	GROM MSGO	189.62 **1.6757e−05**	190.04 **1.6471e−04**	0.30596 **1.5956e−04**	–
F6	GROM MSGO	0.14347 **1.2137e−06**	0.203 **7.4185e−06**	0.064899 **4.9651e−06**	–
F7	GROM MSGO	1.9492e−5 **3.1579e−07**	5.4613e−5 **1.4191e−06**	1.1139e−6 **8.6997e−07**	–
F8	GROM MSGO	- 55179 −8.3797e+04	- 53752 −8.3797e+04	1010.1 1.1073e−04	–
F9	GROM MSGO	**0** **0**	**0** **0**	**0** **0**	≈
F10	GROM MSGO	**8.8817e−16** **8.8818e−16**	**8.8817e−16** **8.8818e−16**	**0** **0**	≈
F11	GROM MSGO	**0** **0**	**0** **0**	**0** **0**	≈
F12	GROM MSGO	1.4254e−5 **2.4721e−12**	5.3593e−5 **4.4801e−09**	7.2927e−5 **6.8312e−09**	–
F13	GROM MSGO	0.09547 **1.9150e−10**	1.0157 **2.9396e−07**	1.4678 **3.3600e−07**	–
Total +					00
Total −					10
Total ≈					03

" −", "+ " and "≈" denote that performance is worse, better and similar to MSGO respectively.

seen that for a single objective problem, the optimal solution non-existent. Hence, finding an optimal solution or even a near-optimal solution is not an easy task. Meta-heuristic optimization algorithms are widely used to solve such kind of problems as they are population-based and start with a randomized solution and derivative-free in nature. Social Group Optimization (SGO)

Table 19
Results on 200 dimensional Benchmark functions with pop_size 40 and 500 iterations.

Function no.	Algorithms	Best	Mean	Std	WRS results
F1	GROM MSGO	3.1809e−65 **0**	5.3021e−65 **0**	1.6155e−65 **0**	−
F2	GROM MSGO	3.7697e−36 **0**	5.2158e−36 **0**	1.0254e−36 **0**	−
F3	GROM MSGO	1.1586e−62 **0**	2.8633e−57 **0**	2.553e−57 **0**	−
F4	GROM MSGO	3.8682e−30 **0**	4.0382e−30 **0**	1.4237e−31 **0**	−
F5	GROM MSGO	195.44 **4.0493e−05**	197.6 **0.0461**	0.55796 **0.0538**	−
F6	GROM MSGO	16.756 **7.2772e−05**	17.942 **0.0049**	1.1032 **0.0047**	−
F7	GROM MSGO	1.7976e−4 **4.9703e−07**	2.8908e−4 **7.2154e−06**	1.1115e−4 **6.2741e−06**	−
F8	GROM MSGO	- 15765 **−8.3797e+04**	- 15167 **−8.3797e+04**	430.67 **0.0177**	−
F9	GROM MSGO	**0** **0**	**0** **0**	**0** **0**	≈
F10	GROM MSGO	**8.8818e−16** **8.8818e−16**	**8.8818e−16** **8.8818e−16**	**0** **0**	≈
F11	GROM MSGO	**0** **0**	**0** **0**	**0** **0**	≈
F12	GROM MSGO	0.15496 **9.1254e−07**	0.1663 **3.3845e−05**	0.00805 **3.4825e−05**	−
F13	GROM MSGO	12.803 **1.2534e−05**	12.822 **4.7378e−04**	0.024253 **5.6438e−04**	−
Total +					00
Total −					10
Total ≈					03

" −", "+ " and "≈" denote that performance is worse, better and similar to MSGO respectively.

Table 20
Optimal costs obtained by different algorithms for case 1.

Algorithms	Best fuel cost per day	Median fuel cost per ay	Mean fuel cost per day	Worse fuel cost per day	Standard deviation
MSGO	**9.391802e+05**	**9.472336e+05**	**9.560759e+05**	1.104793e+06	3.233046e+04
GA-MPC	9.495004e+05	9.696048e+05	9.712892e+05	**9.950425e+05**	1.038739e+04
CDELS	9.830175e+05	NC	1.018834e+06	1.018834e+06	NC
DE–RHC	9.51e+05	NC	1.09e+06	1.42e+06	1.56e+05
RCEP-GA	9.3952e+05	NC	9.5705e+05	1.2204e+06	NC
EA-DE-MA	1.21632e+06	1.56474e+06	1.66379e+06	2.34766e+06	3.27744E+05
SACWIDE	9.64e+5	NC	1.06e+6	1.22e+6	7.53e+4
MSBX_AM	975665.157011	NC	1.098682e+06	1.915280.e+06	NC
CS	9.52e+05	NC	9.52e+05	NC	NC
ABC	1.82e+07	NC	2.26e+07	NC	NC
EPSDE	3.24e+06	NC	6.06e+06	NC	NC

Table 21
Optimal costs obtained by different algorithms for case 2.

Algorithms	Best fuel cost per day	Median fuel cost per day	Mean fuel cost per day	Worse fuel cost per day	Standard deviation
MSGO	1.260586e+06	1.481341e+06	1.543958e+06	2.049798e+06	2.226947e+05
GA-MPC	**9.721018e+05**	**1.049727e+06**	**1.056295e+06**	**1.207998e+06**	5.704156e+04
CDELS	1.459303e+06	NC	1.459303e+06	1.872274e+06	NC
DE–RHC	1.21e+06	NC	1.50e+06	2.13e+06	2.35e+05
RCE-GA	1.1815e+06	NC	1.4409e+06	1.4409e+06	NC
EA-DE-MA	1.4389e+06	2.24644e+06	2.35142e+06	3.77107e+06	4.54677E+05
SACWIDE	9.76e+5	NC	1.14e+6	1.51e+6	1.49e+5
MSBX_AM	1252888.917406	NC	1.508501e+06	1.745728e+06	NC
CS	1.19e+06	NC	1.40e+06	NC	NC
ABC	9.64e+06	NC	1.92e+07	NC	NC
EPSDE	4.47e+06	NC	7.16e+06	NC	NC

Table 22
Optimal costs obtained by different algorithms for case 3.

Algorithms	Best fuel cost per day	Median fuel cost per day	Mean fuel cost per day	Worse fuel cost per day	Standard deviation
MSGO	**9.393650e+05**	**9.491784e+05**	**9.741399e+05**	1.217071e+06	
GA-MPC	9.4659781e+05	9.7575801e+05	9.7510901e+05	**9.9500745e+05**	1.1848936e+04
CDELS	1.002620e+06	NC	1.029531e+06	1.275597e+06	NC
DE–RHC	9.51e+05	NC	1.05e+06	1.43e+06	1.39e+05
RCEP-GA	9.3952e+05	NC	9.7684e+05	1.2204e+06	NC
EA-DE-MA	1.21632e+06	1.56474e+06	1.66379e+06	2.34766e+06	3.27744E+05
SACWIDE	9.55e+5	NC	1.08e+6	1.27e+6	7.04e+4
MSBX_AM	975665.157011	NC	1.098682e+06	1915280.297477	NC
CS	9.56e+05	NC	9.83e+05	NC	NC
ABC	1.85e+07	NC	2.35e+07	NC	NC
EPSDE	4.16e+06	NC	6.21e+06	NC	NC

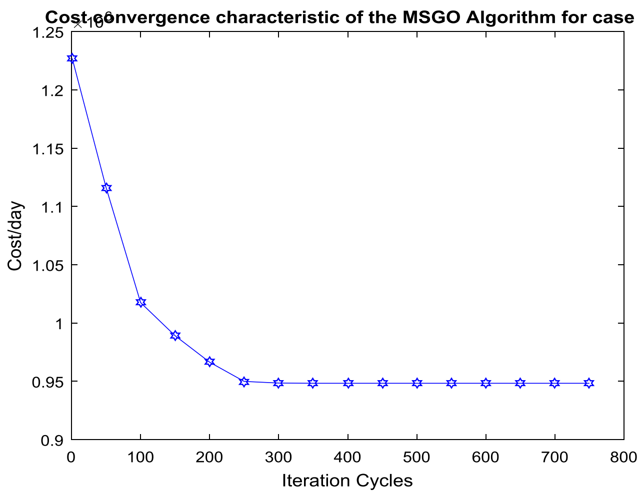

Fig. 3. Cost convergence characteristic of MSGO for case 1.

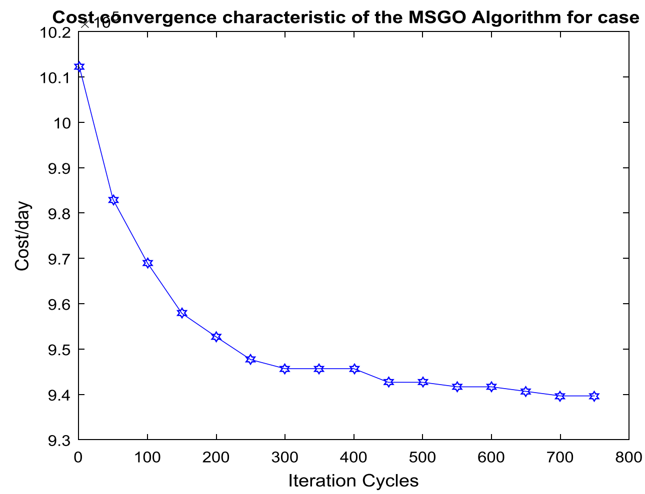

Fig. 5. Cost convergence characteristic of MSGO for case 3.

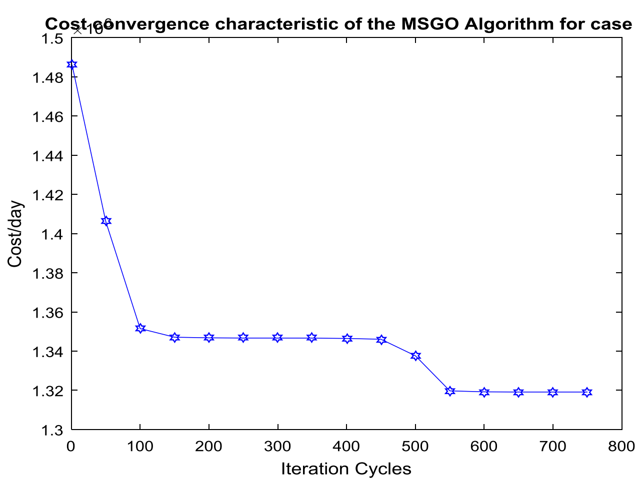

Fig. 4. Cost convergence characteristic of MSGO for case 2.

is a class of meta-heuristic optimization algorithm inspired by the social behavior of an individual in a group to solve such kind of problems. The algorithm finds an optimal solution using two phases in its structure known as improving and acquiring phase. Though SGO found outclass many algorithms in obtaining the optimal solution, it still has shown its deficiency in furthering its ability to explore and exploit its search space. This paper improved the performance of SGO by modifying its acquiring phase with the introduction of a self-awareness probability factor/term. The modified SGO is known as MSGO in this work. To prove the superiority of MSGO in achieving improved exploration and exploitation capabilities, an exhaustive comparative analysis of the performance of MSGO over GA, PSO, DE, ABC, and 21 more recent meta-heuristic algorithms developed from 2010–2019 are presented simulating various types of benchmark functions such as unimodal, multimodal, etc. The results are tabulated in six experiments. Later, MSGO is applied to solve the short-term hydrothermal generation scheduling (HTS) problem. From various tables in the paper and figures, a comprehensive analysis of results is made to show that MSGO is a very promising meta-heuristic algorithm in the family of recent algorithms to solve various complex optimization problems. Even it has shown improved results for the HTS problem, which is highly complex in nature. One of the best things we notice that the MSGO does not deteriorate as the dimension of the problem increases, and at the same time, it can reach optimum points faster in most cases as compared to other studied algorithms in this work. As further research, we would like to ascertain the effectiveness and efficacy of MSGO in solving multi-objective optimization problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Anima Naik ensures that the descriptions are accurate and agreed upon by all authors.

Table A.1
Unimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i-1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	[−100.100]	0
$F_4(x) = \max_i \lvert x_i \rvert, 1 \le i \le n$	30	[−100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30,30]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30	[−1.28,1.28]	0

Table A.2
Multi-modal benchmark functions.

Function	Dim	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500,500]	0
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12,5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[−32, 32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600,600]	0
$F_{12}(x) = \frac{\pi}{n} 10\sin(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2 + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	30	[−50,50]	0
$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]	0

Appendix A

See Tables A.1–A.3.

Appendix B

Codes used in this paper are obtained from below addresses:
ASO: https://ww2.mathworks.cn/matlabcentral/fileexchange/67011 -atom-search-optimization-aso-algorithm
GSA: http://www.mathworks.com/matlabcentral/fileexchange/2775 6-gravitational-search-algorithm-gsa
LSA: https://ww2.mathworks.cn/matlabcentral/fileexchange/54181 -lightning-search-algorithm-lsa
MVO: http://www.alimirjalili.com/MVO.html
SSA: http://www.alimirjalili.com/SSA.html
CSA: https://ww2.mathworks.cn/matlabcentral/fileexchange/56127 -crow-search-algorithm
GWO: https://ww2.mathworks.cn/matlabcentral/fileexchange/44974 -grey-wolf-optimizer-gwo?s_tid=srchtitle.
WOA: http://www.alimirjalili.com/WOA.html
SCA: http://www.alimirjalili.com/SCA.html
HHO: https://www.researchgate.net/publication/331486928_Matlab_code_of_Harris_Hawks_Optimization_HHO
DA: http://www.alimirjalili.com/DA.html
ALO: http://www.alimirjalili.com/ALO.html
LAPO: https://in.mathworks.com/matlabcentral/fileexchange/64459-lapo
GOA: http://www.alimirjalili.com/GOA.html

Abbreviations

Pop_size = Population size
Max_FEs = Maximum number of Function Evaluations
Function Evaluations = FEs
Wilcoxon's Rank-Sum test = WRS test

Table A.3
Fixed-Dimensional multimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	[−65,65]	1
$F_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[−5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	2	[−5,5]	0.398
$F_{18}(x) = [1+(x_1 + x_2 + 1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 x(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_1 \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_1 \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	−3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	(0.10)	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	(0.10)	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	(0.10)	−10.5363

References

[1] M. Črepinšek, S.S. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Comput. Surv. 45 (3) (2013) http://dx.doi.org/10.1145/2480741.2480752.

[2] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, Evol. Comput. IEEE Trans. 1 (1) (1997) 67–82.

[3] S. Satapathy, A. Naik, Social group optimization (SGO) a new population evolutionary optimization technique, Complex Intel. Syst. 2 (3) (2016) 173–203.

[4] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Technical Report, December, 2010.

[5] M.H. Lin, J.F. Tsai, C.S. Yu, A Review of Deterministic Optimization Methods in Engineering and Management, Hindawi Publishing Corporation Mathematical Problems in Engineering, 2012, p. 15, http://dx.doi.org/10.1155/2012/756023, Article ID 756023.

[6] S. Luke, Essentials of Metaheuristics, second ed., 2013, http://cs.gmu.edu/~sean/book/metaheuristics/.

[7] E.G. Talbi, Metaheuristics: From Design to Implementation, vol. 74, John Wiley & Sons, 2009.

[8] C.A. Coello Coello, An Introduction to Evolutionary Algorithms and their Applications, in: Advanced Distributed Systems, Lecture Notes in Computer Science, 3563, Springer Berlin, Heidelberg, 2005, pp. 425–442, http://dx.doi.org/10.1007/11533962_39.

[9] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press Santa Fe, USA, 1999.

[10] A.E. Hassanien, E. Emary, Swarm Intelligence: Principles, Advances, and Applications, CRC Press, 2016.

[11] J.V. Nickerson, Human-Based Evolutionary Computing, in: Handbook of Human Computation, Springer, New York, 2013, pp. 641–648, http://dx.doi.org/10.1007/978-1-4614-8806-4_51.

[12] B. Vahidi, A.F. Nematolahi, Physical and physic-chemical based optimization methods: A review, Soft. Comput. Civ. Eng. (2020) http://dx.doi.org/10.22115/Scce.2020.214959.1161.

[13] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (1998) 95–99.

[14] I. Rechenberg, Evolution Strategy: Optimization of Technical Systems By Means of Biological Evolution, Fromman-Holzboog, Stuttgart, 1973.

[15] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, Evol. Comput. IEEE Trans. 3 (1999) 82–102.

[16] C. Ferreira, Gene expression programming: A new adaptive algorithm for solving problems, Complex Syst. 13 (2) (2001) 87–129.

[17] J.R. Koza, J.P. Rice, Genetic Programming: The Movie, MIT Press, Cambridge, MA, 1992.

[18] N. Hansen, S. Kern, Evaluating the CMA Evolution Strategy on Multimodal Test Functions, Springer, 2004, pp. 282–291.

[19] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (2004) 341–359.

[20] D. Simon, Biogeography-based optimization, Evol. Comput. IEEE Trans. 12 (2008) 702–713.

[21] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, 2008, pp. 1942–1948.

[22] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. 1 (2006) 28–39.

[23] B. Basturk, D. Karaboga, An artificial bee colony (ABC) algorithm for numeric function optimization, in: Proceedings of the IEEE Swarm Intelligence Symposium, 2006, pp. 12–14.

[24] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. 22 (3) (2002) 52–67.

[25] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: Proceedings of the Workshop on Nature Inspired Cooperative Strategies for Optimization, Springer, 2010, pp. 65–74.

[26] X.S. Yang, Firefly algorithm: stochastic test functions and design optimization, Int. J. Bio-Inspired Comput. 2 (2010) 78–84.

[27] A.H. Gandomi, A.H. Alavi, Krill Herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.

[28] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of the World Congress on Nature & Biologically Inspired Computing, 2009, pp. 210–214.

[29] A. Mucherino, O. Seref, Monkey search: a novel metaheuristic search for global optimization, in: AIP Conference Proceedings, 2007, pp. 162–173.

[30] D. Teodorović, M. Dell'Orco, Bee colony optimization–a cooperative learning approach to complex transportation problems, in: Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini–EURO Conference and 10th Meeting of EWGT, Poznan, Publishing House of the Polish Operational and System Research, 2005, pp. 51–60.

[31] S.A. Chu, P.W. Tsai, J.S. Pan, Cat swarm optimization, Lecture Notes in Comput. Sci. 4099 (2006) 854–858.

[32] R. Tang, S. Fong, X.S. Yang, S. Deb, Wolf search algorithm with ephemeral memory, in: Digital Information Management (ICDIM), Seventh International Conference, 2012, pp. 165–172.

[33] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98, http://dx.doi.org/10.1016/j.advengsoft.2015.01.010.

[34] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. (2015) http://dx.doi.org/10.1007/s00521-015-1920-1.

[35] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[36] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[37] A. Alireza, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[38] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, F. Shahrzad, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. (2017) 1–29.
[39] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimization algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[40] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. (2018) http://dx.doi.org/10.1007/s00500-018-3102-4.
[41] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.
[42] A. Heidari, S. Mirjalili, H. Faris, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. (2019) http://dx.doi.org/10.1016/j.future.2019.02.028.
[43] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput.-Aided Des. 43 (2011) 303–315.
[44] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2001) 60–68.
[45] D. Fogel, Artificial Intelligence Through Simulated Evolution, Wiley-IEEE Press, 2009.
[46] F. Glover, Tabu search - Part I, ORSA J. Comput. (1989) 190–206.
[47] F. Glover, Tabu search –Part II, ORSA J. Comput. 2 (1990) 4–32.
[48] S. He, Q. Wu, J. Saunders, A novel group search optimizer inspired by animal behavioral ecology, in: Proceedings of the 2006 IEEE Congress on Evolutionary Computation CEC, 2006, pp. 1272–1278.
[49] S. He, Q.H. Wu, J. Saunders, Group search optimizer: an optimization algorithm inspired by animal searching behaviour, IEEE Trans. Evol. Comput. 13 (5) (2009) 973–990, http://dx.doi.org/10.1109/TEVC.2009.2011992.
[50] G.E. Atashpaz, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: Proceedings of the 2007 IEEE Congress on Evolutionary Computation CEC, 2007, pp. 4661–4667.
[51] A.H. Kashan, League championship algorithm: a new algorithm for numerical function optimization, in: Proceedings of the International Conference on Soft Computing and Pattern Recognition, SOCPAR, 2009, pp. 43–48.
[52] Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in: Advances in Swarm Intelligence, Springer, 2010, pp. 355–364.
[53] A. Kaveh, Colliding bodies optimization, in: Advances in Metaheuristic Algorithms for Optimal Design of Structures, Springer, 2014, pp. 195–232.
[54] A.H. Gandomi, Interior search algorithm (ISA) a novel approach for global optimization, ISA Trans. 53 (4) (2014) 1168–1183, http://dx.doi.org/10.1016/j.isatra.2014.03.018.
[55] A. Sadollah, A. Bahreininejad, H. Eskanda, M. Hamdi, Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (2013) 2592–2612.
[56] N. Moosavian, B. Roodsari, Soccer league competition algorithm, a new method for solving systems of nonlinear equations, Int. J. Intell. Sci. 4 (1) (2014) 7–16, http://dx.doi.org/10.4236/ijis.2014.41002.
[57] C. Dai, Y. Zhu, W. Chen, Seeker optimization algorithm, in: Computational Intelligence and Security, Springer, 2007, pp. 167–176.
[58] F. Ramezani, S. Lotfi, Social-based algorithm (SBA), Appl. Soft Comput. 13 (2013) 2837–2856.
[59] N. Ghorbani, E. Babaei, Exchange market algorithm, Appl. Soft Comput. 19 (2014) 177–187.
[60] M.A. Eita, M.M. Fahmy, Group counseling optimization, Appl. Soft Comput. 22 (2014) 585–604.
[61] M.A. Eita, M.M. Fahmy, Group counseling optimization: a novel approach, in: Research and Development in Intelligent Systems XXVI, Springer, London, 2010, pp. 195–208.
[62] Y. Xu, Z. Cui, J. Zeng, Social emotional optimization algorithm for nonlinear constrained optimization problems, in: Swarm, Evolutionary, and Memetic Computing, Springer, 2010, pp. 583–590.
[63] T.T. Huan, A.J. Kulkarni, J. Kanesan, Ideology algorithm: a socio-inspired optimization methodology, Neural Comput. Appl. (2016) 1–32, http://dx.doi.org/10.1007/s00521-016-2379-4.
[64] Z.Z. Liu, D.H. Chu, C. Song, X. Xue, B.Y. Lu, Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition, Inform. Sci. 326 (2016) 315–333.
[65] A. Naik, S.C. Satapathy, A.S. Ashour, N. Dey, Social group optimization for global optimization of multimodal functions and data clustering problems, Neural Comput. Appl. 30 (1) (2018) 271–287.
[66] H. Emami, F. Derakhshan, Election algorithm: a new socio-politically inspired strategy, AI Commun. 28 (3) (2015) 591–603.
[67] H.C. Kuo, C.H. Lin, Cultural evolution algorithm for global optimizations and its applications, J. Appl. Res. Technol. 11 (4) (2013) 510–522.
[68] A.J. Kulkarni, I.P. Durugkar, M. Kumar, Cohort intelligence: a self supervised learning behavior, in: Systems, Man, and Cybernetics, SMC, IEEE International Conference, IEEE Manchester UK, 2013, pp. 1396–1400.
[69] A.A. Javid, Anarchic society optimization: A human-inspired method, in: Evolutionary Computation, CEC 2011 IEEE Congress, IEEE, New Orleans USA, 2011.
[70] R. Moghdani, K. Salimifard, Volleyball premier league algorithm, Appl. Soft Comput. B64 (2018) 161–185.
[71] M. Kumar, A.J. Kulkarni, S.C. Satapathy, Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology, Future Gener. Comput. Syst. (2017) http://dx.doi.org/10.1016/j.future.2017.10.052.
[72] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simmulated annealing, Science 220 (1983) 671–680.
[73] B. Webster, P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation, in: Proceedings of the 2003 International Conference on Information and Knowledge Engineering, 2003, pp. 255–261.
[74] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, Adv. Eng. Softw. 7 (2006) 106–111.
[75] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (2009) 2232–2248.
[76] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta 213 (2010) 267–289.
[77] R.A. Formato, Central force optimization: A new metaheuristic with applications in applied electromagnetic, Prog. Electromag. Res. 77 (2007) 425–491.
[78] B. Alatas, ACROA: Artificial chemical reaction optimization algorithm for global optimization, Expert. Syst. Appl. 38 (2011) 13170–13180.
[79] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184.
[80] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[81] H. Du, X. Wu, J. Zhuang, Small-world optimization algorithm for function optimization, in: Advances in Natural Computation, Springer, 2006, pp. 264–273.
[82] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, Int. J. Comput. Sci. Eng. 6 (2011) 132–140.
[83] F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, Curved space optimization: A random search based on general relativity theory, 2012, arXiv:1208.2214.
[84] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. (2012) http://dx.doi.org/10.1016/j.compstruc.2012.07.010.
[85] K. Tamura, K. Yasuda, Spiral dynamics inspired optimization, J. Adv. Comput. Intell. Intell. Inf. 15 (8) (2011) 1116–1122.
[86] P. Rabanal, I.R. Fernando Rubio, Using river formation dynamics to design heuristic algorithms, Unconv. Comput. (2007) 163–177.
[87] H. Shareef, A.A. Ibrahim, A.H. Mutlag, Lightning search algorithm, Appl. Soft Comput. J. 36 (2015) 315–333, http://dx.doi.org/10.1016/j.asoc.2015.07.028.
[88] S. Mirjalili, SCA: A Sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2006) 120–133.
[89] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. (2016) http://dx.doi.org/10.1007/s00521-015-1870-7.
[90] A. Foroughi Nematollahi, A. Rahiminejad, B. Vahidi, A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization, Appl. Soft Comput. 59 (2017) 596–621.
[91] W. Zhao, L. Wang, Z. Zhang, A novel atom search optimization for dispersion coefficient estimation in groundwater, Futur. Gener. Comput. Syst. 91 (2019) 601–610, http://dx.doi.org/10.1016/j.future.2018.05.037.
[92] A.F. Nematollahi, A. Rahiminejad, B. Vahidi, A novel meta-heuristic optimization method based on golden ratio in nature, Soft Comput. 24 (2020) 1117–1151, http://dx.doi.org/10.1007/s00500-019-03949-w.
[93] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proc. IEEE Congr. Evol. Comput., 1998, pp. 69–73.
[94] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[95] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proc. IEEE Congr. Evol. Comput. Honolulu, HI, 2002, pp. 1671–1676.

[96] K.E. Parsopoulos, M.N. Vrahatis, UPSO—A Unified particle swarm optimization scheme, in: Lecture Series on Computational Sciences, 2004, pp. 868–873.

[97] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: Simpler, maybe better, IEEE Trans. Evol. Comput. 8 (2004) 204–210.

[98] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Proc. Swarm Intelligence Symp., 2003, pp. 174–181.

[99] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295, http://dx.doi.org/10.1109/TEVC.2005.857610.

[100] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.

[101] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.

[102] J. Zhang, A.C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[103] R. Mallipeddi, P.N. Suganthan, Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies, in: Swarm, Evolutionary, and Memetic Computing, in: Lecture Notes in Computer Science, vol. 6466, Springer, Berlin, Heidelberg, 2010.

[104] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector, 15, (1) 2011, pp. 55–66, http://dx.doi.org/10.1109/TEVC.2010.2087271.

[105] S.C. Satapathy, A. Naik, Improved teaching learning based optimization for global function optimization, Int. J. Decis. Sci. Lett. 2 (1) (2012) 23–34.

[106] S.C. Satapathy, A. Naik, K. Parvathi, A teaching learning based optimization based on orthogonal design for solving global optimization problems, SpringerPlus 2 (130) (2013) 1–12.

[107] S.C. Satapathy, A. Naik, K. Parvathi, Weighted teaching learning based optimization for global function optimization, Appl. Math. 4 (2013) 429–439.

[108] S.C. Satapathy, A. Naik, A modified teaching learning based optimization (mTLBO) for global search, Recent Patent Comput. Sci. 6 (1) (2013) 60–72.

[109] S.C. Satapathy, A. Naik, Modified teaching -learning-based optimization algorithm for global numerical optimization-A comparative study, Swarm Evol. Comput. (2014) http://dx.doi.org/10.1016/j.swevo.2013.12.005i.

[110] S.C. Satapathy, A. Naik, Cooperative teaching–Learning based optimisation for global function optimisation, Int. J. Appl. Res. Inf. Technol. Comput. (IJARITAC) 4 (1) (2013) 1–17.

[111] A. Naik, Use of teaching learning based optimization for data clustering, Innov. Electr. Power Eng. Commun. Comput. Technol. 360 (2020) 123–134.

[112] N. Dey, V. Rajinikanth, A. Ashour, J.M. Tavares, Social group optimization supported segmentation and evaluation of skin melanoma images, 10, (2) 2018, p. 51.

[113] V. Rajinikanth, S.C. Satapathy, Segmentation of ischemic stroke lesion in brain MRI based on social group optimization and fuzzy-tsallis entropy, 43, (8) 2018, pp. 4365–4378.

[114] J. Fang, H. Zheng, J. Liu, J. Zhao, Y. Zhang, K. Wang, A transformer fault diagnosis model using an optimal hybrid dissolved gas analysis features subset with improved social group optimization-support vector machine classifier energies, 11, (8) 2018, p. 1922.

[115] M.T. Tran, H.A. Pham, V.L. Nguyen, A.T. Trinh, Optimisation of stiffeners for maximum fundamental frequency of cross-ply laminated cylindrical panels using social group optimisation and smeared stiffener method, Thin-Walled Struct. 120 (2017) 172–179.

[116] S.P. Praveen, K.T. Rao, B. Janakiramaiah, Effective allocation of resources and task scheduling in cloud environment using social group optimization, Arab. J. Sci. Technol. 43 (8) (2018) 4265–4272.

[117] V.S. Chakravarthy, P.S.R. Chowdary, S.C. Satapathy, Antenna array synthesis using social group optimization, 471, 2018, pp. 895–905.

[118] G. Madhavi, V. Harika, Implementation of social group optimization to economic load dispatch problem, Int. J. Appl. Eng. Res. 13 (2018) 11195–11200.

[119] R. Monisha, R. Mrinalini, M.N. Britto, Social group optimization and Shannon's function-based RGB image multi-level thresholding, Smart Intell. Comput. Appl. 105 (2019) 123–132.

[120] P. Parwekar, SGO: A new approach for energy efficient clustering in WSN, in: Sensor Technology: Concepts, Methodologies, Tools, and Applications, 2018, http://dx.doi.org/10.4018/978-1-7998-2454-1.ch034.

[121] A.E. Eiben, C.A. Schippers, On evolutionary exploration and exploitation, Fundam. Inform. 35 (1998) 35–50.

[122] S. Kumar, R. Naresh, Efficient real coded genetic algorithm to solve the nonconvex hydrothermal scheduling problem, Int. J. Electr. Power Energy Syst. 29 (10) (2007) 738–747.

[123] K.P. Wong, Y.W. Wong, Short-term hydrothermal scheduling part. I. Simulated annealing approach, IEEE Proc: Gener. Transm. Distrib. 141 (5) (1994) 497–501.

[124] X. Bai, S.M. Shahidehpour, Hydro-thermal scheduling by tabu search and decomposition method, IEEE Trans. Power Syst. 11 (2) (1996) 968–974.

[125] S.J. Huang, Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches, IEEE Trans. Energy Convers. 16 (3) (2001) 296–301.

[126] K.K. Mandal, M. Basu, N. Chakraborty, Particle swarm optimization technique based short-term hydrothermal scheduling, Appl. Soft Comput. 8 (4) (2008) 1392–1399.

[127] P.K. Hota, A.K. Barisal, R. Chakrabarti, An improved PSO technique for short-term optimal hydrothermal scheduling, Electr. Power Syst. Res. 79 (7) (2009) 1047–1053.

[128] S. Lu, C. Sun, Z. Lu, An improved quantum-behaved particle swarm optimization method for short-term combined economic emission hydrothermal scheduling, Energy Convers. Manage 51 (3) (2010) 561–571.

[129] M. Basu, Improved differential evolution for short-term hydrothermal scheduling, Int. J. Electr. Power Energy Syst. 58 (2014) 91–100.

[130] Y. Wang, J. Zhou, L. Mo, R. Zhang, Y. Zhang, Short-term hydrothermal generation scheduling using differential real-coded quantum-inspired evolutionary algorithm, Energy 44 (1) (2012) 657–671.

[131] X. Liao, J. Zhou, S. Ouyang, R. Zhang, Y. Zhang, An adaptive chaotic artificial bee colony algorithm for short-term hydrothermal generation scheduling, Int. J. Electr. Power Energy Syst. 53 (2013) 34–42.

[132] M. Basu, An interactive fuzzy satisfying method based on evolutionary programming technique for multiobjective short-term hydrothermal scheduling, Electr. Power Syst. Res. 69 (2–3) (2004) 277–285, http://dx.doi.org/10.1016/j.epsr.2003.10.003.

[133] S. Sivasubramani, K. Shanti Swarup, Hybrid DE–SQP algorithm for non-convex short term hydrothermal scheduling problem, Energy Convers. Manage. 52 (1) (2011) 757–761.

[134] S.M. Elsayed, R.A. Sarker, D.L. Essam, GA with a New Multi-Parent Crossover for Solving IEEE-CEC2011 Competition Problems, 2011, 978-1-4244-7835-4/11/$26.00 ©2011 IEEE.

[135] A. Mandal, A.K. Das, P. Mukherjee, S. Das, P.N. Suganthan, Modified Differential Evolution with Local Search Algorithm for Real World Optimization, 2011, 978-1-4244-7835-4/11/$26.00 ©2011 IEEE.

[136] A. LaTorre, S. Muelas, J.M. Datsi, F. Informática, U.P. Madrid, Benchmarking a Hybrid DE-RHC Algorithm on Real World Problems, 2011, 978-1-4244-7835-4/11/$26.00 ©2011 IEEE.

[137] A. Saha, T. Ray, How does the good old Genetic Algorithm fare at Real World Optimization, 2011, 978-1-4244-7835-4/11/$26.00 ©2011 IEEE.

[138] K.H. Singh, T. Ray, Performance of a Hybrid EA-DE-Memetic Algorithm on CEC 2011 Real World Optimization Problems, 2011, 978-1-4244-7835-4/11/$26.00 ©2011 IEEE.

[139] U. Halder, S. Das, D. Maity, A. Abraham, P. Dasgupta, Self Adaptive Cluster Based and Weed Inspired Differential Evolution Algorithm For Real World Optimization, 2011, 978-1-4244-7833-0/11/$26.00 ©2011 IEEE.

[140] S. Bandaru, R. Tulshyan, K. Deb, Modified SBX and Adaptive Mutation for Real World Single Objective Optimization, 2011, 978-1-4244-7835-4/11/$26.00 ©2011 IEEE.

[141] M. Essaidy, L. Idoumghary, J. Lepagnoty, M. Brévilliersy, D. Fodereanx, A hybrid differential evolution algorithm for real world problems, in: Conference Paper, 2018, http://dx.doi.org/10.1109/CEC.2018.847796.

[142] M.P. Fay, M.A. Proschan, Wilcoxon–Mann–Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules, Stat. Surv. 4 (2010) 1–39, http://dx.doi.org/10.1214/09-SS051.

[143] M.Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.

[144] M.Y. Cheng, L.C. Lien, Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization, J. Comput. Civ. Eng. 26 (5) (2012) 612–624.

[145] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (1) (2008) 687–697.

[146] T. Krink, B. Filipic, G.B. Fogel, Noisy optimization problems-a particular challenge for differential evolution, in: Congress on Evolutionary Computation, CEC2004 IEEE, vol. 1, 2004, pp. 332–339.

[147] S. Surjanovic, D. Bingham, British Columbia, 2015, https://www.sfu.ca/~ssurjano/optimization.htm.