Memetic Comp.
DOI 10.1007/s12293-016-0212-3

REGULAR RESEARCH PAPER

Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems

Gai-Ge Wang[1,2,3,4]

Received: 5 May 2016 / Accepted: 23 August 2016

Abstract Phototaxis, signifying movement of an organism towards or away from a source of light, is one of the most representative features for moths. It has recently been shown that one of the characteristics of moths has been the propensity to follow Lévy flights. Inspired by the phototaxis and Lévy flights of the moths, a new kind of metaheuristic algorithm, called moth search (MS) algorithm, is developed in the present work. In nature, moths are a family insects associated with butterflies belonging to the order Lepidoptera. In MS method, the best moth individual is viewed as the light source. Some moths that are close to the fittest one always display an inclination to fly around their own positions in the form of Lévy flights. On the contrary, due to phototaxis, the moths that are comparatively far from the fittest one will tend to fly towards the best one directly in a big step. These two features correspond to the processes of exploitation and exploration of any metaheuristic optimization method. The phototaxis and Lévy flights of the moths can be used to build up a general-purpose optimization method. In order to demonstrate the superiority of its performance, the MS method is further compared with five other state-of-the-art metaheuristic optimization algorithms through an array of experiments on fourteen basic benchmarks, eleven IEEE CEC 2005 complicated benchmarks and seven IEEE CEC 2011 real world problems. The results clearly demonstrate that MS significantly outperforms five other methods on most test functions and engineering cases.

Keywords Metaheuristic algorithms · Moth search algorithm · Swarm intelligence · Lévy flights · Benchmark functions · Real world problems · Performance analysis

1 Introduction

Nowadays, most engineering problems are turning out to be too complicated for the traditional methods to provide solutions that are accurate or even satisfactory. However, modern nature-inspired metaheuristic algorithms have been found to be effective in coming up with optimal or near-optimal solutions for these types of NP-hard problems, such as data encryption [1], and image processing [2]. Among those, the swarm-based algorithms and evolutionary algorithms (EAs) have emerged as the most representative stochastic paradigms.

The idea of swarm intelligence (SI) originates from the collective behavior of scattered, self-organized systems. Its incorporation into the optimization methods led to the arrival of the so-called swarm-based algorithms, popularly known as SI methods. These SI methods have been successfully used to address various important application problems, such as path planning, navigation [3,4], and big data optimization [5,6]. Particle swarm optimization (PSO) [7] and ant colony optimization (ACO) [8] are two of the most representative paradigms within the entire set of SI methods. These two are respectively inspired by the flocking behavior of birds and the pheromone of ants, and had since been followed by more and

✉ Gai-Ge Wang
gaigewang@gmail.com; gaigewang@163.com

1 School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, Jiangsu, China

2 Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada

3 Institute of Algorithm and Big Data Analysis, Northeast Normal University, Changchun 130117, China

4 School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

more effective SI methods, e.g., artificial bee colony (ABC) [9,10], monarch butterfly optimization (MBO) [11], elephant herding optimization (EHO) [12], and krill herd (KH) [13,14]. The inspiration behind those originates from the smarm behavior of honey bees, and krill, respectively. Individuals in swarm intelligence are synonymous to memes in the context of memetic computing [15,16].

On the other hand, motivated by the genetic evolution process and natural selection, several evolutionary algorithms (EAs) have been proposed by various researchers. One of the earliest among the EAs is genetic algorithms (GAs). In general, GA, evolutionary programming (EP), genetic programming (GP) and evolution strategy (ES) are clearly four of the most representative EAs among the entire family. In addition, differential evolution (DE) [17,18] is an efficient EA that has been widely used in an innumerable number of applications. Stud genetic algorithm (SGA) [19] is an enhanced version of GA, where only the best individual is selected to generate offspring. Biogeography-based optimization (BBO), based on the rule of biogeography [20], has recently been developed by Simon. By incorporating multi-stage strategy into the basic GP, multi-stage genetic programming (MSGP) [21] was propounded for the purpose of non-linear system modeling. Animal migration optimization (AMO) [22] has been proposed by Li et al. and is inspired by animal migration behavior. By simulating two ways of reproduction, a new methodology, called earthworm optimization algorithm (EWA) [23], intended for addressing global optimization, has recently been studied by Wang et al. [23].

We commence by reviewing the phototaxis and Levy flights of moths that is followed by their generalization in order to formulate a general-purpose metaheuristic method. The outcome is a new kind of metaheuristic algorithm, called moth search (MS) algorithm, which is presented in this paper. In the first place, the moths having smaller distance from the best one are always flying around their own positions in the form of Lévy flights. On the other hand, the best moth, considered to be the target for the moths, is situated at a distance from the rest. It will lead to their flying towards the best one in line on account of phototaxis. The process mentioned above can be considered as exploitation and exploration of an optimization algorithm, i.e., local search and global search. The act of balancing of exploitation and exploration is indeed, a demanding one and the fine balance itself is a tough optimization problem often regarded as an optimization problem of an optimization algorithm. In order to demonstrate its performance and effectiveness, the MS method has been benchmarked by twenty-five standard test functions and seven real world problems in comparison with five other state-of-the-art metaheuristic algorithms. It has been experimentally proved that MS significantly outperforms five other methods on most test functions and engineer cases. Hence MS method is a successful and effective supplement for SI portfolio.

The remainder of this paper is organized as follows. Section 2 reviews the phototaxis and the flight behavior of moths in nature, while Sect. 3 discusses how the phototaxis and flight behavior of moths can be utilized towards formulating a general-purpose search heuristic. In order to demonstrate the performance of the MS, various experimental results, comparing MS with other five optimization methods, both for general benchmark functions and for seven engineering case studies, are presented in Sect. 4. The paper ends with Sect. 5 after presenting some concluding remarks and sharing the views of the authors vis-a-vis suggestions for further work.

2 The behavior of moths

Moths, related to butterflies, are a kind of insects that belong to the order Lepidoptera. In the world, the total species of moth stand nearly about 160,000 with majority of them being nocturnal. Among the various characteristics of moths, the phototaxis and Levy flights are two of the most representative features as described as follows.

2.1 Phototaxis

In general, moths tend to fly around the light source, phenomena known as phototaxis. Hitherto the exact reason for phototaxis remains unknown. There are several hypotheses to explain this phenomenon. One of those is that celestial navigation is used in transverse orientation while flying. The moths will fly in a straight line so as to remain at a fixed angle to the celestial light, like the moon. In fact, the angle between moth and the light source keeps on changing, but it often escapes from observation, because celestial objects are so far away. On the other hand, if a moth that is close to the light source uses it for navigation, the changes in angle will be clearly discernible, even from a small distance. The moth will instinctively tend to do its best to adjust the flight orientation so as to move towards the light source all the time, causing the airborne moths to come plummeting downward. This will lead to a spiral flight path that gets closer and closer to the light source [24].

2.2 Lévy flights

Non-Gaussian, heavy-tailed statistics is becoming a commonly used tool in several applications, such as the behavior of numerous animals and insects. As a kind of random movements, Lévy flights are one of the most important flight patterns in natural environments. Many species, such as Drosophila, have been shown to fly in the form of Lévy flights that can be approximated to be power law distributed over a range of scales with the feature of exponents close to 3/2 [25]. Reynolds et al. use sex pheromone as source in order to study the flight patterns of moth [26]. The observed results indicate that some of the complex flight patterns that arise in the experiments have proven to be in compliance with the usage of an optimal biased scale-free (Lévy-flights) searching technique [26].

Lévy flights or anomalous diffusion processes describe a class of random walks, whose step lengths is in the form of a power-law tailed distribution. In general, Lévy flights are a kind of random walks, whose steps are drawn from Lévy distribution. Lévy distribution can be mathematically expressed in the form of a power-law formula, as shown in Eq. (1).

$$L(s) \sim |s|^{-\beta} \tag{1}$$

where $1 < \beta \leq 3$ is an index.

The moth flight patterns as shown in [26] are the initially reported example of Lévy-flight movement patterns with $\beta \approx 1.5$. In addition, Lévy flights can maximize the efficiency of resource search in uncertain environments. Therefore, Lévy flights of moths with $\beta = 1.5$ are used to optimize benchmarks and engineering cases as discussed in the following section.

Because Lévy flights can significantly improve the search ability of metaheuristic algorithms, it has been combined with many state-of-the-art algorithms, such as CS [27], KH, and FA.

3 Moth search algorithm

This section intends to describe the process of the optimization tasks by using the phototaxis and Lévy flights of moths.

3.1 Lévy flights

The moths, having a smaller distance from the best one, will fly around the best one in the form of Lévy flights. In other words, their positions are updated by performing Lévy flights, as described in Eq. (2).

For moth i, it can be updated as:

$$x_i^{t+1} = x_i^t + \alpha L(s) \tag{2}$$

where x_i^{t+1} and x_i^t are respectively the updated and original position at generation t, and t is the current generation. $L(s)$ is the step drawn from Lévy flights. The parameter α is the scale factor related to the problem of interest. In our current work, it can be given as:

$$\alpha = S_{max}/t^2 \tag{3}$$

where $S_{\max}$ is max walk step and its value is set according to the given problem.

Lévy distribution $L(s)$ in Eq. (1) can be formulated as follows:

$$L(s) = \frac{(\beta - 1)\Gamma(\beta - 1)\sin\left(\frac{\pi(\beta-1)}{2}\right)}{\pi s^{\beta}} \tag{4}$$

where s is bigger than 0. $\Gamma(x)$ is the gamma function.

As aforementioned, the Lévy flights of moths can be drawn from the Lévy distribution with $\beta = 1.5$. Therefore, in the present paper, β is set to 1.5 for our experiments.

3.2 Fly straightly

Certain moths that are distant from the light source will fly towards that source of light in line. This process can be described below.

For moth i, its flights can be formulated as

$$x_i^{t+1} = \lambda \times \left(x_i^t + \phi \times \left(x_{best}^t\text{-}x_i^t\right)\right) \tag{5}$$

where x_{best}^t is the best moth at generation t, and φ is an acceleration factor that is set to golden ration in our present work. λ is a scale factor.

On the other hand, the moth may fly towards the final position that is beyond the light source. For this case, the final position for moth i can be formulated as:

$$x_i^{t+1} = \lambda \times \left(x_i^t + \frac{1}{\varphi} \times \left(x_{best}^t\text{-}x_i^t\right)\right) \tag{6}$$

For simplicity, for moth i, its position will be updated by Eq. (5) or Eq. (6) with the possibility of 50 %. Moreover, these two updating process mentioned above can be represented in Fig. 1a, b, respectively. In Fig. 1, x_{best}, x_i, and $x_{i,new}$ are respectively the best, original and updated position for moth i, and they are considered as a light source, start point, and end point. λ is a scale factor which can control the convergence speed of the algorithm and improve the diversity of the population. In our current work, the scale factor is set to a random number drawn by the standard uniform distribution.

3.3 Schematic description of MS algorithm

In MS method, for simplicity, the whole moth population is divided into two equal Subpopulations (Subpopulation 1 and Subpopulation 2) according to their fitness, and they are updated by Sects. 3.1 and 3.2, respectively. This is tantamount to say that the moths in Subpopulation 1 are much closer to the best one than that of Subpopulation 2.

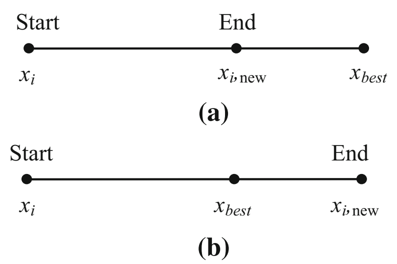


Fig. 1 A simplified representation of the rectilinear flight with **a** x_{best} (*right*) and **b** x_{best} (*middle*)

In addition, like many other metaheuristic algorithms, an elitism strategy is incorporated into the MS method with the aim of accelerating the convergence of the MS method.

According to the analyses above, the phototaxis and Lévy flights of moths can be used to construct a general-purpose metaheuristic algorithm, and moth search (MS) algorithm has been designed, whose main description can be given in Algorithm 1 and Fig. 2. In Algorithm 1, *MaxGen* is the maximum generation that can be considered as the termination condition.

4 Simulation results

In this section, MS method is fully investigated from various respects through serials of experiments on twenty-five benchmarks (F01–F25) and seven real world problems (RWP01–RWP07). For benchmark cases, functions F01–F14 are the basic benchmarks, and F15–F25 are rotated, shifted, and composition CEC 2005 benchmarks (see Table 1). Each function has twenty variables, i.e., the dimension of test functions is twenty. More information on all the benchmarks can be found in [20].

In order to decrease the influence of randomness, fifty independent implementations are performed under the same conditions as shown in [28]. This will also yield fair comparisons among different metaheuristic algorithms.

For MS method, its parameters are set as indicated below:

The number of kept moths at each generation $= 2$, the index $\beta = 1.5$, max walk step $S_{\max} = 1.0$, acceleration factor $\varphi = (5^{1/2} - 1)/2 \cong 0.618$. All the methods use a population of 50 individuals (that is, the population size $NP = 50$). For other parameters used in different methods, their settings can be referred to [29]. In the following experiments, Function Evaluations (FEs) is considered as termination condition and is set to 10^4 if there is no special clarification.

Algorithm 1: Moth search algorithm

Begin

Step 1: Initialization. Set the generation number t=1; randomly initialize the population P of *NP* moths randomly using uniform distribution; the maximum generation *MaxGen*, max walk step $S_{\max}$, the index β, and acceleration factor φ.

Step 2: Fitness evaluation. Evaluate each moth individual according to its position.

Step 3: While t<*MaxGen* **do**

Sort all the moth individuals as per their fitness.

for i= 1 to *NP*/2 (for all moth individuals in Subpopulation 1) **do**

Generate x_i^{t+1} by performing Lévy flights as Section 3.1;

end for i

for i= *NP*/2+1 to *NP* (for all moth individuals in Subpopulation 2) **do**

if *rand*>0.5 **then**

Generate x_i^{t+1} by Eq. (5);

else

Generate x_i^{t+1} by Eq. (6).

end if

end for i

Evaluate the population as per the newly updated positions;

$t = t$+1.

Step 4: end while

Step 5: Output the best solution.

End.

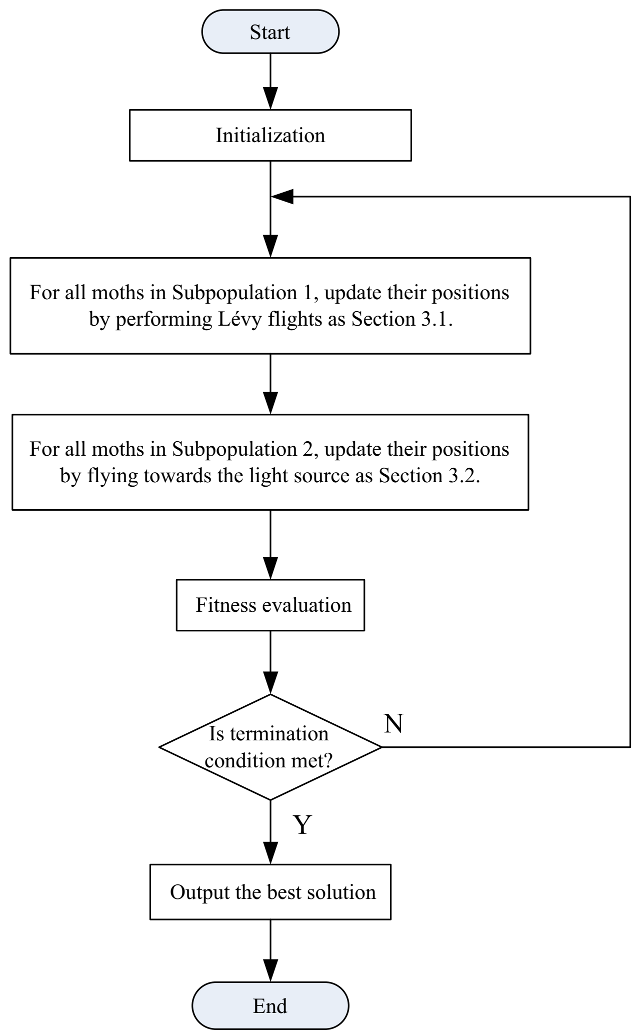


Fig. 2 Schematic flowchart of MS method

4.1 Comparisons of the optimal MS with other metaheuristic algorithms

In this section, the performance of MS method will be fully exploited on 25 benchmarks in comparison with five *state-of-the-art* metaheuristic algorithms (ABC [9], BBO [20], DE [17], PSO [7], and SGA [19]). The results obtained are recorded in Tables 2, 3, 4 and 5.

From Table 2, it is evident that, on average, MS significantly outperforms other five methods on 21 out of 25 benchmarks. BBO and SGA rank two and three, and they can find the fittest solutions on three (F03, F09, and F24) and one (F25) out of twenty-five functions, respectively.

For the best solutions, Table 3 shows that MS method has the best performance on eighteen out of 25 benchmarks. BBO and SGA are able to find the fittest solutions on five (F09, F11, F13, F23, and F24) and four (F13, F21, F22 and F25) out of 25 benchmarks. Generally, ABC, DE, and PSO have identical performance each other. A careful observation of Table 3 reveals that BBO, MS and SGA have the same performance on function F13, which is much better than ABC, DE, and PSO.

For the worst function values shown in Table 4, clearly, MS method outperforms five other methods, and it performs the best on eighteen out of 25 benchmarks. BBO and SGA rank the second and the third, and they both are well capable of searching for the fittest solutions on four (F03, F09, F15, and F20) and two (F24 and F25) out of 25 functions respectively. In addition, DE can find the final best solution on function F17.

Furthermore, Table 4 reveals that MS method can find the best solutions with the smallest standard deviation (Std) on fourteen out of 25 benchmarks among six comparative methods. In other words, the solutions obtained by MS method

Table 1 Benchmark functions

No.	Name	No.	Name
F01	Ackley	F14	Zakharov
F02	Dixon & Price	F15	Shifted Rotated Griewank's Function without Bounds
F03	Fletcher-Powell	F16	Shifted Rotated Ackley's Function with Global Optimum on Bounds
F04	Griewank	F17	Shifted Rastrigin's Function
F05	Pathological function	F18	Shifted Rotated Weierstrass Function
F06	Penalty #1	F19	Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)
F07	Penalty #2	F20	Shifted Rotated Expanded Scaffer's F6
F08	Perm	F21	Rotated Hybrid Composition Function
F09	Schwefel 2.26	F22	Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum
F10	Schwefel 1.2	F23	Rotated Hybrid Composition Function with the Global Optimum on the Bounds
F11	Schwefel 2.22	F24	Composition Function with High Condition Number Matrix
F12	Schwefel 2.21	F25	Rotated Hybrid Composition Function
F13	Step		

Table 2 Mean function values

	ABC	BBO	DE	MS	PSO	SGA
F01	16.45	3.77	18.26	**2.4E−6**	18.44	4.33
F02	4.6E7	7.7E4	3.8E6	**0.67**	1.4E7	1.1E4
F03	2.7E5	**7.0E4**	2.5E5	1.6E5	5.0E5	8.3E4
F04	85.88	3.33	21.42	**1.00**	73.02	2.19
F05	6.91	4.44	3.71	**2.2E−16**	4.63	4.30
F06	2.0E7	3.49	3.0E5	**0.06**	4.8E6	1.28
F07	5.1E7	3.5E3	2.8E6	**1.07**	2.6E7	6.47
F08	1.2E51	6.1E51	4.5E45	**2.5E37**	4.5E47	6.0E51
F09	3.3E3	**376.85**	3.4E3	5.0E3	4.7E3	383.63
F10	2.5E4	8.5E3	2.7E4	**5.4E−11**	1.9E4	1.2E4
F11	31.04	0.86	22.50	**4.9E−6**	71.08	5.20
F12	66.07	38.52	52.84	**2.0E−6**	47.28	40.00
F13	35.68	1.16	9.26	**1.00**	27.50	1.44
F14	231.34	101.05	238.22	**8.3E−11**	199.24	177.23
F15	4.5E3	2.7E3	2.8E3	**2.6E3**	5.1E3	3.0E3
F16	21.03	20.97	21.02	**20.67**	20.97	21.01
F17	168.86	200.03	125.39	**120.96**	232.09	199.98
F18	115.33	116.91	117.37	**110.58**	116.61	117.76
F19	119.10	17.13	22.92	**12.52**	70.53	16.57
F20	9.03	8.73	9.11	**8.67**	9.05	8.71
F21	1.2E3	1.1E3	1.1E3	**910.00**	1.2E3	1.0E3
F22	1.2E3	1.0E3	1.1E3	**910.00**	1.2E3	1.0E3
F23	1.2E3	1.0E3	1.1E3	**910.00**	1.3E3	1.1E3
F24	1.5E3	**1.3E3**	1.3E3	1.4E3	1.5E3	1.3E3
F25	1.6E3	1.1E3	1.3E3	1.5E3	1.6E3	**985.39**
Total	0	3	0	21	0	1

Table 3 Best function values

	ABC	BBO	DE	MS	PSO	SGA
F01	13.35	2.51	16.48	**2.1E−8**	17.05	2.51
F02	4.0E6	4.6E3	1.2E6	**0.67**	4.1E6	2.1E3
F03	1.2E5	3.7E4	1.7E5	**3.4E4**	3.3E5	3.9E4
F04	30.93	1.79	10.96	**1.00**	34.86	1.37
F05	5.87	3.47	2.03	**2.2E−16**	1.04	3.01
F06	8.1E4	0.99	60.54	**0.01**	6.7E5	0.16
F07	6.8E6	2.80	2.3E5	**0.40**	3.2E6	2.10
F08	1.4E45	6.0E51	3.7E37	**3.0E32**	3.7E43	6.0E51
F09	2.4E3	**132.16**	2.5E3	4.0E3	2.9E3	133.03
F10	1.8E4	2.0E3	1.5E4	**2.2E−14**	1.1E4	3.4E3
F11	20.12	**2.2E−16**	14.93	1.3E−7	28.14	1.00
F12	55.00	22.00	37.39	**9.4E−8**	33.69	13.00
F13	16.00	**1.00**	6.00	**1.00**	20.00	**1.00**
F14	137.57	39.00	151.77	**7.6E−16**	92.98	68.31
F15	2.1E3	1.6E3	1.4E3	**1.3E3**	2.7E3	1.9E3
F16	20.81	20.54	20.82	**20.39**	20.79	20.90
F17	78.63	199.93	97.20	**76.77**	178.06	199.93
F18	109.30	112.96	115.07	**104.77**	112.40	114.09
F19	29.41	14.30	17.35	**5.32**	26.90	14.30
F20	8.71	8.06	8.61	**7.82**	8.59	7.87
F21	1.1E3	731.50	848.84	910.00	1.1E3	**648.81**
F22	990.59	872.24	957.05	910.00	1.0E3	**676.83**
F23	1.1E3	**735.63**	895.20	910.00	1.1E3	859.99
F24	1.3E3	**1.2E3**	1.2E3	1.3E3	1.3E3	1.2E3
F25	1.5E3	669.98	1.1E3	1.4E3	1.4E3	**655.72**
Total	0	5	0	18	0	4

will fall into the smallest scope. BBO and SGA have the smallest Std values on four and three out of 25 benchmarks, respectively, while DE and PSO exhibit stable performance on two out of twenty-five benchmarks.

From Tables 2, 3, 4 and 5, it can be opined that the MS method can locate the fittest solutions with the smallest Std values for most cases. This indicates that under the same conditions, MS algorithm is one of the most suitable methods, when searching for the best solutions in most cases.

With the aim of clearly showing the effectiveness of MS, the results obtained by three most representative methods are illustrated in this section (Fig. 3).

It can be observed from Fig. 3 that, for 16 functions (F01, F02, F05–F08, F10, F12, F14, F16–F18 and F20–F23), MS method is far better than BBO and SGA. Although BBO, MS and SGA have identical final solutions on four functions (F04, F11, F13 and F15), MS has a much faster convergent speed than BBO and SGA for these cases, whereas MS shows worse performance than BBO and SGA on four functions (F03, F09, F24 and F25). In addition, the three methods exhibit similar performance on function F19 and their convergent curves are hard to be differentiated each other. Figure 3 clearly indicates that MS has a significant advantage over BBO and SGA on most benchmarks.

4.2 Comparisons using *t*-test

According to the results obtained by six methods on 25 functions, a two-tailed t-test is performed on each function with the 5 % level of significance between MS and other five methods. The t values are recorded in Table 6. When comparing with other methods, the values are highlighted in **Boldface** if MS exhibits significantly superior performance than the compared algorithm. For the last three rows, the “Better”, “Equal”, and “Worse” indicate that MS is better than, equal to, and worse than the compared methods. For example, for the comparisons between BBO and MS, MS can find better, equal, and worse solutions on eighteen, three, and four benchmarks respectively. That is tantamount to saying that MS performs better than BBO on most test cases. In addition, in consideration of MS and SGA, the numbers in last three rows are 20, 1 and 4, respectively. This indicates that

Table 4 Worst function values

	ABC	BBO	DE	MS	PSO	SGA
F01	17.85	5.77	18.97	**1.3E−5**	18.88	6.42
F02	1.0E8	2.8E5	9.0E6	**0.67**	3.2E7	4.0E4
F03	4.0E5	**1.2E5**	3.3E5	3.9E5	8.2E5	1.8E5
F04	136.66	5.83	31.13	**1.00**	104.06	3.98
F05	7.44	5.98	4.70	**2.2E−16**	6.34	5.54
F06	5.5E7	6.85	1.6E6	**0.14**	1.4E7	2.70
F07	1.4E8	6.3E4	9.9E6	**1.38**	7.8E7	110.00
F08	6.0E51	1.0E52	6.0E46	**3.5E38**	3.1E48	6.0E51
F09	3.8E3	**651.39**	4.1E3	6.2E3	5.8E3	753.27
F10	3.2E4	1.6E4	3.3E4	**6.5E10**	2.7E4	2.4E4
F11	41.35	3.00	30.02	**2.6E−5**	86.00	9.00
F12	74.00	65.00	65.50	**8.7E−6**	66.00	73.00
F13	49.00	2.00	14.00	**1.00**	36.00	4.00
F14	306.26	167.00	320.38	**4.9E−10**	333.35	305.00
F15	7.2E3	**4.1E3**	5.0E3	4.9E3	8.5E3	5.0E3
F16	21.18	21.12	21.13	**20.95**	21.13	21.13
F17	236.63	200.89	**142.71**	153.33	267.94	201.27
F18	117.80	118.75	118.99	**115.61**	118.55	119.73
F19	310.29	39.34	35.24	**30.09**	164.38	44.62
F20	9.37	**9.17**	9.33	9.23	9.27	9.22
F21	1.4E3	1.4E3	1.4E3	**910.00**	1.6E3	1.3E3
F22	1.4E3	1.2E3	1.5E3	**910.00**	1.5E3	1.4E3
F23	1.6E3	1.3E3	1.6E3	**910.00**	1.5E3	1.4E3
F24	2.0E3	1.6E3	1.7E3	1.7E3	1.8E3	**1.6E3**
F25	1.8E3	1.6E3	1.6E3	1.7E3	1.9E3	**1.6E3**
Total	0	4	1	18	0	2

Table 5 The Std of different methods

	ABC	BBO	DE	MS	PSO	SGA
F01	0.91	0.66	0.45	**2.6E−6**	0.44	0.83
F02	2.4E7	6.7E4	1.7E6	**5.7E−5**	5.8E6	8.2E3
F03	7.6E4	**2.0E4**	4.0E4	7.0E4	1.1E5	3.0E4
F04	24.39	0.92	5.29	**3.6E−15**	12.52	0.58
F05	0.35	0.45	0.61	**0.00**	1.07	0.52
F06	1.4E7	1.35	3.1E5	**0.02**	3.1E6	0.58
F07	2.9E7	1.3E4	2.1E6	**0.19**	1.4E7	15.04
F08	1.6E51	5.7E50	1.1E46	5.6E37	6.9E47	**1.3E36**
F09	255.96	**121.30**	356.50	425.70	741.40	136.99
F10	3.5E3	3.4E3	4.3E3	**1.1E−10**	4.8E3	3.9E3
F11	4.69	0.83	2.96	**5.2E−6**	10.72	2.60
F12	4.97	10.24	5.58	**2.1E−6**	6.39	11.72
F13	6.76	0.37	1.71	**0.00**	3.64	0.70
F14	32.91	28.93	42.40	**1.2E−10**	53.47	53.22
F15	1.2E3	**685.76**	818.96	885.15	1.4E3	861.95
F16	0.09	0.10	0.07	0.12	0.07	**0.06**
F17	23.09	0.24	10.57	17.40	19.93	**0.21**
F18	1.49	1.17	**1.08**	2.46	1.29	1.18
F19	64.22	7.52	**3.13**	4.61	32.05	6.83
F20	0.16	0.26	0.16	0.32	**0.15**	0.31
F21	84.39	125.68	102.67	**2.3E−8**	105.09	149.07
F22	88.14	76.98	110.59	**3.4E−8**	102.70	122.50
F23	105.02	98.09	124.83	**4.6E−8**	113.08	106.57
F24	128.44	**84.85**	117.72	91.54	113.24	88.45
F25	89.97	315.40	111.71	87.11	**83.25**	307.96
Total	0	4	2	14	2	3

MS method is superior to or equal to SGA on 21 out of 25 functions, while MS has worse performance than SGA on four out of 25 benchmarks. Although the MS do not take the advantage over the comparative algorithms on all the functions, Table 6 demonstrates that it is superior to the other five algorithms on most functions.

4.3 Real world problems

Except for the standard functions discussed in the sections above, seven more real world problems (RWPs) that are selected from CEC 2011 havealso been used to validate the effectiveness of MS.

4.3.1 RWP01: optimal control of a non-linear stirred tank reactor

A first-order chemical reaction conducted in a continuous stirred tank reactor (CSTR) can be considered a multimodal optimization problem that has been considered standard benchmark to test metaheuristic algorithms. This process can be modeled in mathematical form as shown in Eqs. (7) and (8).

$$\dot{x}_1 = -(2+u)(x_1+0.25)+(x_2+0.5)\exp\left(\frac{25x_1}{x_1+2}\right), \quad (7)$$

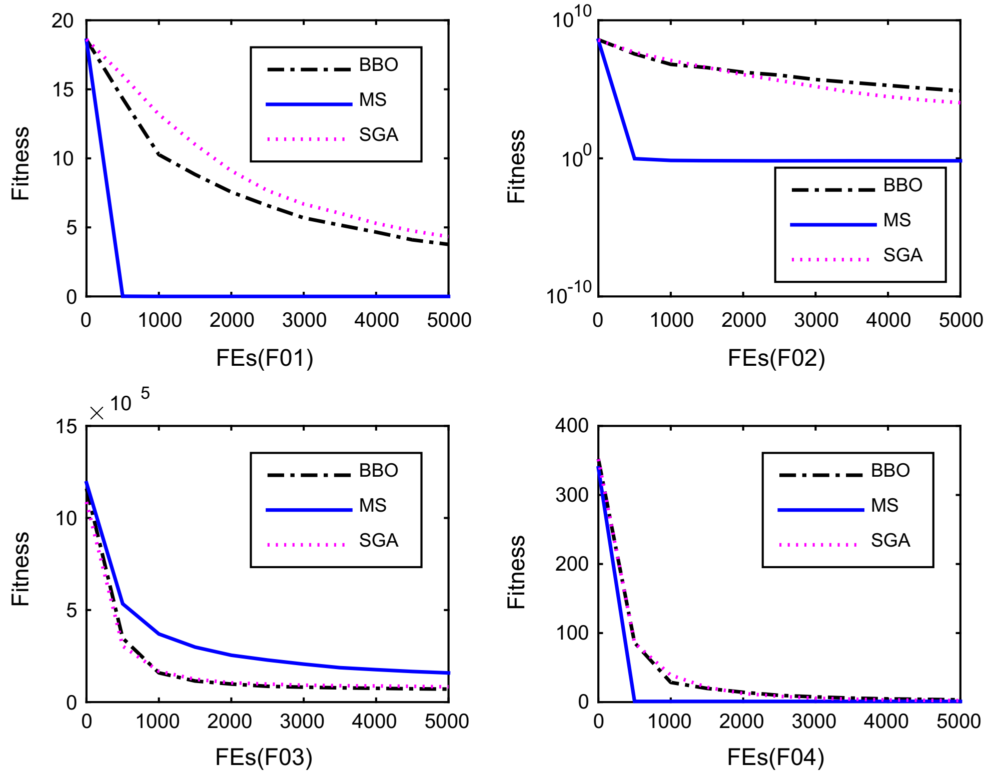
$$\dot{x}_2 = 0.5 - x_2 - (x_2+0.5)\exp\left(\frac{25x_1}{x_1+2}\right), \quad (8)$$

where $u(t)$, x_1 and x_2 are the flow rate of the cooling fluid, state temperature, and deviation, respectively. A suitable value of u will be determined with the aim of optimizing the performance index that can be given as follows:

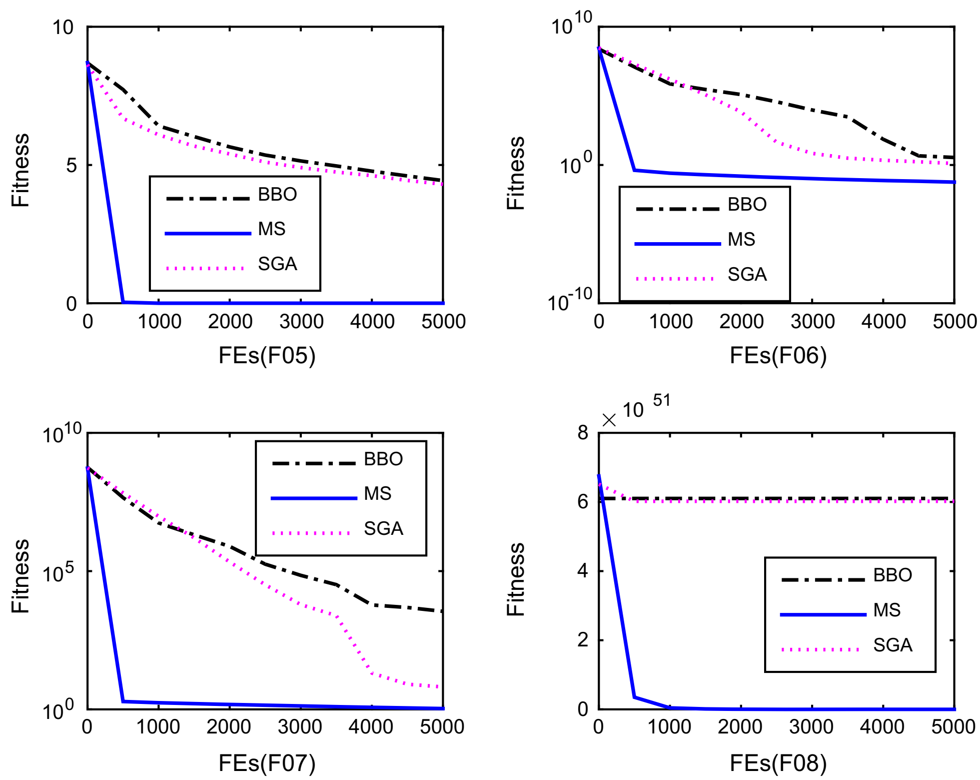
$$J = \int_0^{t_f=0.72} (x_1^2 + x_2^2 + 0.1u^2)dt \quad (9)$$

4.3.2 RWP02: spread spectrum radar polly phase code design

The choice of the appropriate waveform is a critical factor that is worthy of study when designing a radar-system with pulse compression. Among the different radar pulse modulation methods, polyphase codes have been studied in details. A new

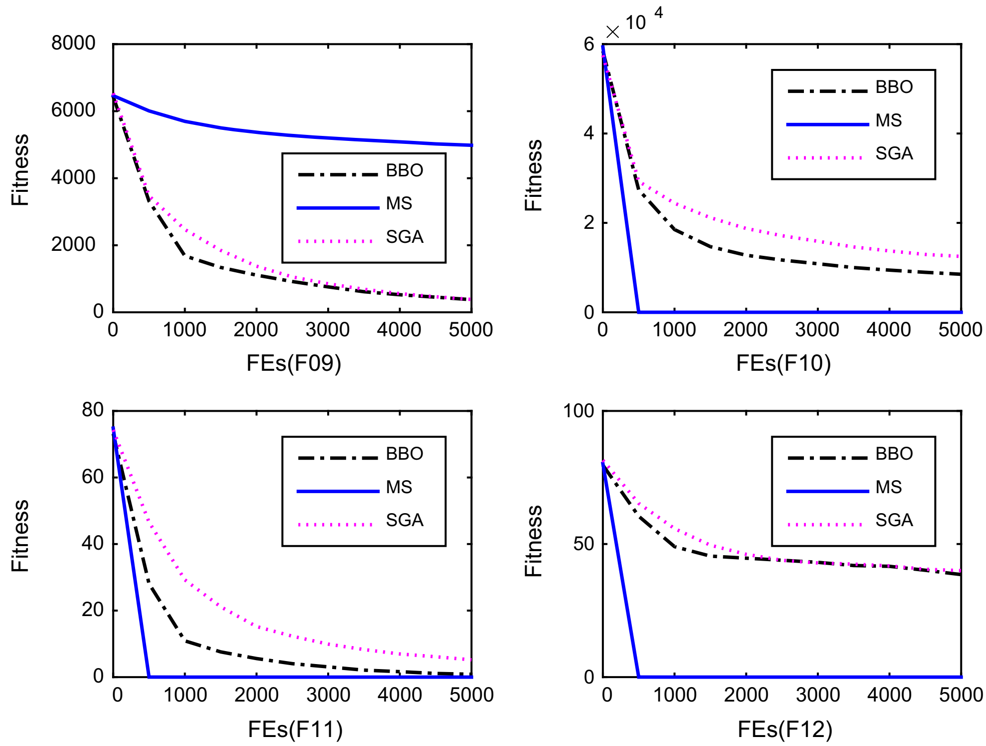


(a)

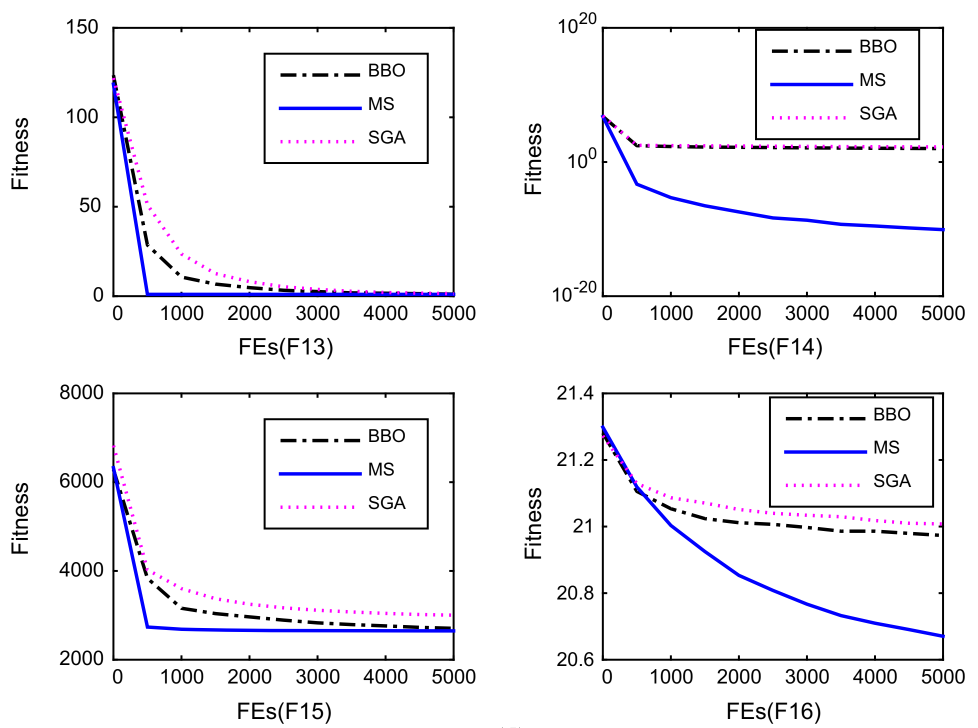


(b)

Fig. 3 Convergent curves of the most preventative methods. **a** F01–F04; **b** F05–F08; **c** F09–F12; **d** F13–F16; **e** F17–F20; **f** F21–F24; **g** F25



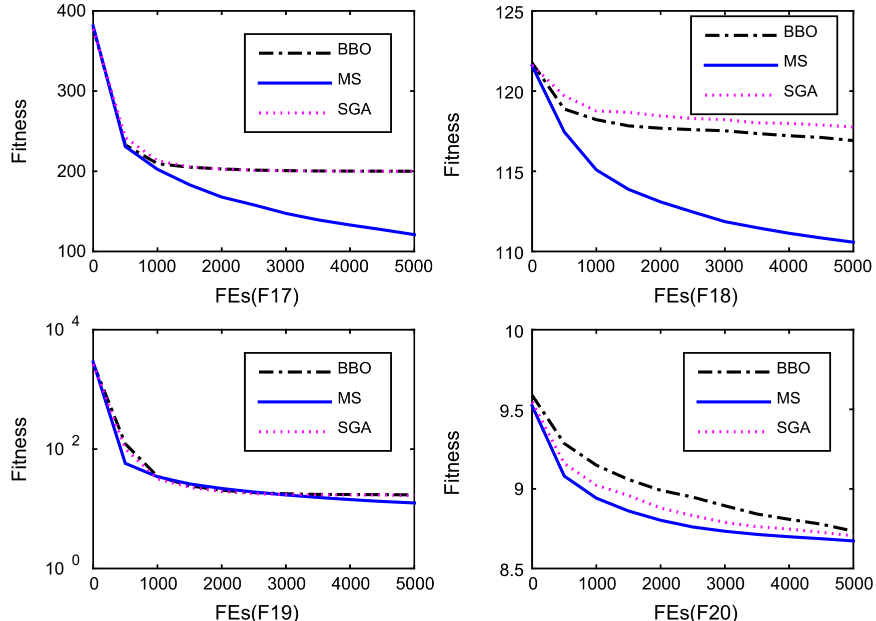
(c)



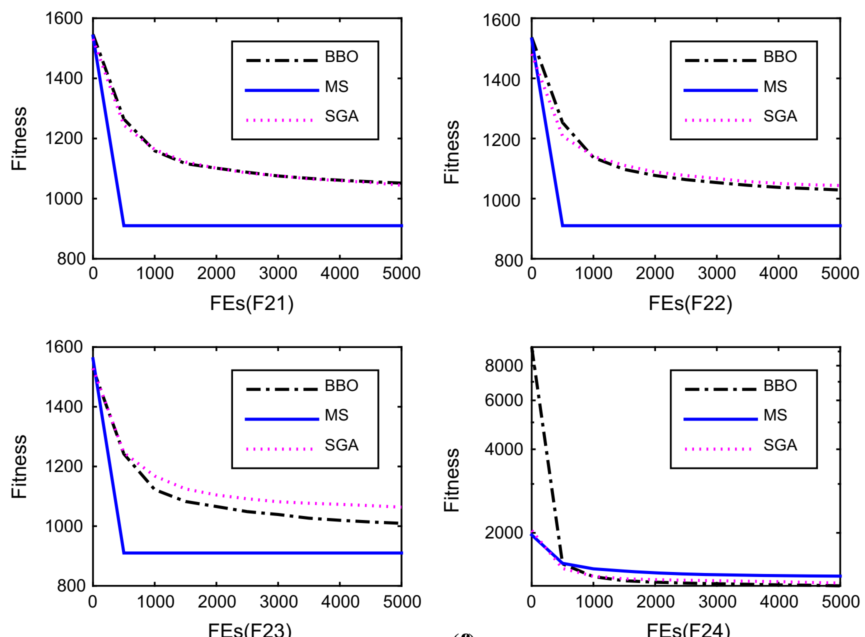
(d)

Fig. 3 continued

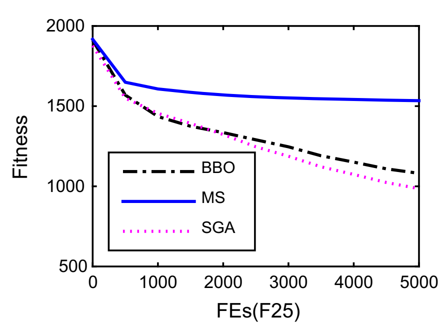
Fig. 3 continued



(e)



(f)



(g)

Table 6 Comparisons between MS and other methods at $\alpha = 0.05$ on a two-tailed t-tests

	ABC	BBO	DE	PSO	SGA
F01	**127.28**	**40.17**	**285.10**	**293.40**	**36.73**
F02	**13.61**	**8.16**	**15.27**	**16.96**	**9.16**
F03	**7.43**	−8.51	**8.02**	**18.73**	−6.99
F04	**24.60**	**17.90**	**27.30**	**40.68**	**14.44**
F05	**140.05**	**69.34**	**43.05**	**30.57**	**58.24**
F06	**9.91**	**18.00**	**6.98**	**10.91**	**14.98**
F07	**12.63**	1.93	**9.65**	**13.21**	**2.54**
F08	**5.28**	**75.04**	**2.86**	**4.62**	**7.6E14**
F09	−23.52	−73.58	−20.31	−2.24	−72.73
F10	**51.05**	**17.89**	**43.59**	**28.22**	**22.46**
F11	**46.77**	**7.30**	**53.72**	**46.87**	**14.17**
F12	**93.97**	**26.59**	**66.92**	**52.31**	**24.13**
F13	**36.26**	**3.06**	**34.12**	**51.51**	**4.42**
F14	**49.71**	**24.70**	**39.73**	**26.35**	**23.55**
F15	**8.86**	0. 36	0. 63	**10.87**	**2.03**
F16	**17.24**	**13.50**	**17.94**	**14.84**	**17.73**
F17	**11.72**	**32.14**	1.54	**29.70**	**32.12**
F18	**11.72**	**16.48**	**17.89**	**15.39**	**18.64**
F19	**11.70**	**3.69**	**13.19**	**12.67**	**3.47**
F20	**6.95**	1.06	**8.68**	**7.54**	0.51
F21	**25.44**	**7.97**	**11.86**	**21.77**	**6.37**
F22	**24.31**	**10.91**	**12.35**	**21.04**	**7.71**
F23	**20.44**	**7.17**	**11.82**	**22.48**	**10.19**
F24	**4.59**	−6.13	−2.94	**3.22**	−4.33
F25	**6.42**	−9.79	−10.66	**5.19**	−12.12
Better	**24**	**18**	**20**	**24**	**20**
Equal	0	3	2	0	1
Worse	1	4	3	1	4

polyphase pulse compression code synthesis method [30] is proposed, which is based on the aperiodic autocorrelation function. The problem can be considered as a continuous optimization problem formulated below:

$$\text{global} \min_{x \in X} f(x) = \max\{\phi_1(x), \ldots, \phi_{2m}(x)\}, \tag{10}$$

where $m = 2n - 1$ and

$$\phi_{2i-1}(x) = \sum_{j=i}^{n} \cos\left(\sum_{k=|2i-j-1|+1}^{j} x_k\right), \quad i = 1, \ldots, n \tag{11}$$

$$\phi_{2i}(x) = 0.5 + \sum_{j=i+1}^{n} \cos\left(\sum_{k=|2i-j|+1}^{j} x_k\right), \quad i = 1, \ldots, n-1 \tag{12}$$

$$\phi_{m+i}(x) = -\phi_i(x), \quad i = 1, \ldots, m \tag{13}$$

The objective function is piecewise smooth NP-hard problem that can be further addressed by metaheuristic algorithms.

4.3.3 RWP03: large scale transmission pricing problem

Recently, transmission pricing is a hotly debated issue in the modern power systems [30]. The transmission pricing generally deals with the issue of assigning the fixed costs of transmission to different take-holders. Its scheme is determined by several factors, and many transmission pricing methods have been proposed.

Among those, equivalent bilateral exchange (EBE) [30] is in the form of the linearized model in the system. The basic EBE method [30] generates a load-generation interaction matrix as per the proportionality principle in Eq. (14):

$$GD_{ij} = \frac{P_{Gi} P_{Dj}}{P_D^{sys}}, \tag{14}$$

where, GD_{ij} is the amount of equivalent bilateral exchange and P_D^{sys} is the total load. The fraction of power flow in line k is due to bilateral exchange is evaluated for all equivalent bilateral power exchanges [30]. The net flow in line k can be expressed as:

$$pf_k = \sum_i \sum_j \left|\gamma_{ij}^k\right| GD_{ij} \tag{15}$$

4.3.4 RWP04: dynamic economic dispatch (DED) problem

The dynamic economic dispatch (DED) problem has the property of the hourly dispatch problem. However, in the present problem, the power demand fluctuates with each passing hour, and the 24-h power generation schedule is to be determined. Therefore, the DED problem can be considered as an optimization task, whose dimension is 24 times that of the static ELD problem.

The objective function related to the production cost can be approximated to be a quadratic function of the active power outputs from the generating units. Therefore, it can be given as:

$$\text{MINIMIZE}: F_c = \sum_{k=1}^{T} \sum_{i=1}^{N_G} F_{ih}(P_{ih}) \tag{16}$$

$$F_{it}(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i, \quad i = 1, 2, 3, \ldots, N_G \tag{17}$$

where $F_{it}(P_{it})$ is the cost function while a_i, b_i and c_i are its cost coefficients, P_{it} is the real power output, N_G is the number of online generating units. The cost function for unit with valve point loading effect can be given as:

Table 7 Optimization results obtained via six methods over 50 runs and 1000 FEs for seven CEC 2011 real world problems

	ABC	BBO	DE	MS	PSO	SGA
RWP01						
BEST	19.81	23.93	20.93	**13.79**	14.39	23.93
MEAN	21.43	23.93	22.22	**18.22**	20.47	23.93
WORST	22.91	23.93	23.93	**21.83**	21.83	23.93
STD	0.51	**7.2E−15**	0.84	3.15	1.78	**7.2E−15**
RWP02						
BEST	1.62	1.73	1.88	**1.45**	1.95	1.62
MEAN	2.34	2.13	2.44	**2.06**	2.42	2.17
WORST	2.88	2.49	2.77	**2.45**	2.89	2.90
STD	0.24	**0.20**	0.22	0.22	0.20	0.30
RWP03						
BEST	1.6E6	**8.9E5**	1.1E6	9.6E5	1.4E6	1.0E6
MEAN	2.0E6	1.2E6	1.3E6	**1.1E6**	1.7E6	1.3E6
WORST	2.6E6	1.5E6	1.5E6	**1.4E6**	2.0E6	1.6E6
STD	1.7E5	1.0E5	9.3E4	**8.7E4**	1.3E5	1.3E5
RWP04						
BEST	7.8E7	5.7E6	3.9E7	**2.1E6**	8.6E7	1.8E7
MEAN	3.3E8	5.2E7	2.8E8	**1.0E7**	1.2E8	8.8E7
WORST	4.0E8	9.2E7	3.3E8	**1.1E7**	1.6E8	1.5E8
STD	3.2E7	2.0E7	2.4E7	**2.6E5**	2.4E7	2.3E7
RWP05						
BEST	2.2E8	1.2E7	2.2E8	**9.7E6**	7.2E7	2.5E7
MEAN	1.5E8	1.6E7	6.2E7	**9.1E6**	1.1E8	4.0E7
WORST	1.9E8	3.6E7	8.5E7	**2.6E7**	1.3E8	5.9E7
STD	2.1E7	6.7E6	1.1E7	**5.8E6**	1.0E7	9.1E6
RWP06						
BEST	8.2E7	**4.6E6**	4.0E7	5.0E6	7.9E7	1.4E7
MEAN	1.5E8	1.7E7	6.2E7	**1.2E7**	1.1E8	3.7E7
WORST	2.0E8	4.0E7	8.5E7	**3.9E7**	1.3E8	5.5E7
STD	2.2E7	8.5E6	1.1E7	**6.4E6**	1.3E7	9.3E6
RWP07						
BEST	1.0E8	2.5E6	3.6E7	**2.3E6**	7.9E7	2.2E7
MEAN	1.5E8	1.4E7	5.9E7	**1.1E7**	1.1E8	3.8E7
WORST	1.9E8	3.2E7	8.1E7	**3.0E7**	1.3E8	6.1E7
STD	2.1E7	**6.5E6**	8.9E6	7.3E6	1.2E7	8.9E6
Total						
BEST	0	2	0	5	0	0
MEAN	0	0	0	7	0	0
WORST	0	0	0	7	0	0
STD	0	3	0	4	0	1

$$F_{it}\left(P_{it}\right) = a_i P_{it}^2 + b_i P_{it} + c_i + \left| e_i \sin\left(f_{it} \left(P_{it}^{\min} - P_{it} \right) \right) \right| \tag{18}$$

where e_i and f_i are the cost coefficients with respect to the valve point loading effect.

4.3.5 RWP05, RWP06 and RWP07: hydrothermal scheduling problem

In a hydrothermal scheduling problem, the objective function is the overall fuel cost of thermal units within the given short period of time. The hydrothermal system discussed in this paper is a rather complicated one, therefore, the global optimum cost is difficult to be discovered by traditional optimization methods.

The system studied here contains a multi-chain cascaded network of four hydro plants and an equivalent thermal power plant. The objective is to maximize the output of the hydro units such that the thermal unit takes up the only minimum load.

The total fuel cost for this system is considered as the objective function F, and can be expressed as:

$$\text{MINIMIZE } F = \sum_{i=1}^{M} f_i \left(P_{Ti}\right) \tag{19}$$

where M is the total number of intervals while f_i represents the cost function associated with the equivalent thermal unit's power generation P_{Ti} at the i-th interval expressed as :

$$f_i \left(P_{Ti}\right) = a_i P_{Ti}^2 + b_i P_{Ti} + c_i + \left| e_i \sin \left(f_i \left(P_{Ti}^{\min} - P_{Ti} \right) \right) \right| \tag{20}$$

In CEC 2011, this problem involves three cases. For a clear description, the three cases correspond to three engineering cases, called RWP05, RWP06, and RWP07.

More detailed information about the above seven RWPs can be found in [30].

4.3.6 Optimization results

Seven RWPs described in Sects. 4.3.1–4.3.5 can be treated as continuous or discrete constrained problems so as to authenticate the performance of various metaheuristic algorithms. In our current work, six methods are used to optimize seven RWPs. For the experiments conducted in this section, population size *NP* is set to 30. Similarly, the fixed FEs are treated as termination condition and it is set to 1000. All the other parameters used here are the same with the description mentioned above. The results obtained by 50 independent runs on seven RWPs are given in Table 7.

MS generates the best solutions on five out of seven RWPs (except RWP03 and RWP06), while for BBO, the optimum solutions can be arrived at from RWP03 and RWP06. For the average solution, MS can come up with the best solutions on all the RWPs. For the worst solution, similar to the average solution, MS can find the best solutions on all the RWPs. For Std of six RWPs, MS have the smallest Std on four RWPs, while BBO and SGA have the smallest Std on three and one out of seven RWPs. The results in Table 7 indicate that MS exhibits the best performance among six methods discussed here, although its Std is only the best on four RWPs.

In addition, running time is one of the most important factors while dealing with practical problems by using a metaheuristic algorithm. Here, the average computational

Table 8 Average computational requirements of six methods for twenty-five benchmarks and seven CEC 2011 real world problems

	ABC	BBO	DE	MS	PSO	SGA
Benchmarks	1.34	2.54	1.81	**1.00**	3.75	4.54
CEC 2011_RWPs	1.08	1.46	1.28	**1.00**	1.69	2.19

requirements of six methods are collected from the experiments mentioned above, and they are provided in Table 8. The results in Table 8 imply that MS is the fastest method and uses the least time when confronting 25 benchmarks and CEC 2011 RWPs. BBO is only inferior to MS that consumes more time as compared to the MS method. The results in Tables 2, 3, 4, 5, 6, 7 and 8 demonstrate that MS can find far better solutions with less computational requirements w.r.t. other five comparative methods.

5 Conclusions

In this paper, the phototaxis and Lévy flights of moths in nature are idealized and generalized in order to develop MS, a general-purpose metaheuristic method. In MS method, the best moth at each generation is considered as light source. The moths that have smaller distances from the light source tend to fly around their own positions in the form of Lévy flights. In contrast, the moths that are distant from the light source will fly towards the light source in a straight line. The process mentioned above corresponds to the exploitative and explorative process of an optimization algorithm. In order to fully investigate the performance of MS, it is compared with five *state-of-the-art* metaheuristic algorithms on a total of 25 standard test functions and seven real world problems. It has been clearly demonstrated that MS can search for the best solutions more effectively, efficiently and accurately w.r.t. five other methods on most test functions and engineering applications. In addition, MS does not involve complicated operations, and hence its implementation is both easy and flexible.

Though MS has several advantages as outlined above, the following four points will be addressed in our future research.

Firstly, in this paper, little effort has been made to fine-tune the parameters used in the MS method. Is it better than the current implementations if acceleration factor φ is not set to golden ratio? Additionally, what is the best number of moths in Subpopulation 1 and Subpopulation 2?

Secondly, in our work, only 25 benchmarks and seven engineering cases have been used to test the performance of our proposed MS method. In the future, more benchmarks and especially engineering optimization problems will be used to further verify MS method.

Thirdly, in this paper, the phototaxis and Lévy flights of moths are generalized to develop MS method. In the future, more useful features will be incorporated into the MS method, which will likely enhance its performance to a great extent.

At last, here, the superiority of MS is only experimentally proven. In the future, the convergence of MS will be analyzed in theory and investigated using the theory of dynamic systems and Markov chain.

Acknowledgments This work was supported by Jiangsu Province Science Foundation for Youths (No. BK20150239) and National Natural Science Foundation of China (No. 61503165). The authors thanked Prof. Suash Deb and Prof. Leandro dos Santos Coelho's help for improving language.

References

1. Fu Z, Ren K, Shu J, Sun X, Huang F (2015) Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans Parallel Distrib Syst 1–14. doi:10.1109/tpds.2015.2506573
2. Chen B, Shu H, Coatrieux G, Chen G, Sun X, Coatrieux JL (2015) Color image analysis by quaternion-type moments. J Math Imaging Vis 51(1):124–144. doi:10.1007/s10851-014-0511-6
3. Mohanty PK, Parhi DR (2015) A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach. Memet Comput 7(4):255–273. doi:10.1007/s12293-015-0160-3
4. Kundu S, Parhi DR (2016) Navigation of underwater robot based on dynamically adaptive harmony search algorithm. Memet Comput 8(2):125–146. doi:10.1007/s12293-016-0179-0
5. Elsayed S, Sarker R (2016) Differential evolution framework for big data optimization. Memet Comput 8(1):17–33. doi:10.1007/s12293-015-0174-x
6. Zhang Y, Liu J, Zhou M, Jiang Z (2016) A multi-objective memetic algorithm based on decomposition for big optimization problems. Memet Comput 8(1):45–61. doi:10.1007/s12293-015-0175-9
7. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Paper presented at the Proceeding of the IEEE international conference on neural networks, Perth, Australia, 27 November–1 December
8. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B Cybern 26(1):29–41. doi:10.1109/3477.484436
9. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471. doi:10.1007/s10898-007-9149-x
10. Bullinaria JA, AlYahya K (2014) Artificial Bee Colony training of neural networks: comparison with back-propagation. Memet Comput 6(3):171–182. doi:10.1007/s12293-014-0137-7
11. Wang G-G, Deb S, Cui Z (2015) Monarch butterfly optimization. Neural Comput Appl. doi:10.1007/s00521-015-1923-y
12. Wang G-G, Deb S, Gao X-Z, Coelho LDS (2016) A new metaheuristic optimization algorithm motivated by elephant herding behavior. Int J Bio-Inspired Comput. http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijbic
13. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simulat 17(12):4831–4845. doi:10.1016/j.cnsns.2012.05.010
14. Wang G-G, Gandomi AH, Alavi AH (2014) Stud krill herd algorithm. Neurocomputing 128:363–370. doi:10.1016/j.neucom.2013.08.031
15. Le MN, Ong Y-S, Jin Y, Sendhoff B (2009) Lamarckian memetic algorithms: local optimum and connectivity structure analysis. Memet Comput 1(3):175–190. doi:10.1007/s12293-009-0016-9
16. Meuth R, Lim M-H, Ong Y-S, Wunsch DC (2009) A proposition on memes and meta-memes in computing for higher-order learning. Memet Comput 1(2):85–100. doi:10.1007/s12293-009-0011-1
17. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359. doi:10.1023/A:1008202821328
18. Chong JK (2016) A novel multi-objective memetic algorithm based on opposition-based self-adaptive differential evolution. Memet Comput 8(2):147–165. doi:10.1007/s12293-015-0170-1
19. Khatib W, Fleming P (1998) The stud GA: a mini revolution? In: Eiben A, Back T, Schoenauer M, Schwefel H (eds) Proceedings of the 5th international conference on parallel problem solving from nature, New York, USA, 1998. Parallel problem solving from nature. Springer-Verlag, London, UK, pp 683–691
20. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713. doi:10.1109/TEVC.2008.919004
21. Gandomi AH, Alavi AH (2011) Multi-stage genetic programming: a new strategy to nonlinear system modeling. Inf Sci 181(23):5227–5239. doi:10.1016/j.ins.2011.07.026
22. Li X, Zhang J, Yin M (2014) Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput Appl 24(7–8):1867–1877. doi:10.1007/s00521-013-1433-8
23. Wang G-G, Deb S, Coelho LDS (2015) Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Int J Bio-Inspired Comput. http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijbic
24. Callahan PS (1977) Moth and candle: the candle flame as a sexual mimic of the coded infrared wavelengths from a moth sex scent (pheromone). Appl Opt 16(12):3089–3097. doi:10.1364/AO.16.003089
25. Reynolds AM, Jones HB, Hill JK, Pearson AJ, Wilson K, Wolf S, Lim KS, Reynolds DR, Chapman JW (2015) Evidence for a pervasive 'idling-mode' activity template in flying and pedestrian insects. R Soc Open Sci 2(5):150085. doi:10.1098/rsos.150085
26. Reynolds AM, Reynolds DR, Smith AD, Svensson GP, Lofstedt C (2007) Appetitive flight patterns of male Agrotis segetum moths over landscape scales. J Theor Biol 245(1):141–149. doi:10.1016/j.jtbi.2006.10.007
27. Li X, Yin M (2015) Modified cuckoo search algorithm with self adaptive parameter method. Inf Sci 298:80–97. doi:10.1016/j.ins.2014.11.042
28. Wang G, Guo L, Wang H, Duan H, Liu L, Li J (2014) Incorporating mutation scheme into krill herd algorithm for global numerical optimization. Neural Comput Appl 24(3–4):853–871. doi:10.1007/s00521-012-1304-8
29. Wang G-G, Guo L, Gandomi AH, Hao G-S, Wang H (2014) Chaotic krill herd algorithm. Inf Sci 274:17–34. doi:10.1016/j.ins.2014.02.123
30. Das S, Suganthan P (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur Univ., Nanyang Technol. Univ, Kolkata, India