Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation xxx (xxxx) xxx
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Nature-inspired approach: An enhanced moth swarm algorithm for global optimization

Qifang Luo[a,b], Xiao Yang[a], Yongquan Zhou[a,b,*]

[a] College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China
[b] Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China

Received 5 December 2017; received in revised form 13 October 2018; accepted 31 October 2018
Available online xxxx

Abstract

The moth swarm algorithm (MSA) is a recent swarm intelligence optimization algorithm, but its convergence precision and ability can be limited in some applications. To enhance the MSA's exploration abilities, an enhanced MSA called the elite opposition-based MSA (EOMSA) is proposed. For the EOMSA, an elite opposition-based strategy is used to enhance the diversity of the population and its exploration ability. The EOMSA was validated using 23 benchmark functions and three structure engineering design problems. The results show that the EOMSA can find a more accurate solution than other population-based algorithms, and it also has a fast convergence speed and high degree of stability.

Keywords: Elite opposition-based learning; Enhanced moth swarm algorithm; Function optimization; Structure engineering design; Nature-inspired approach

1. Introduction

From a general point of view, optimization is the process of searching for the global solution of a problem under a given circumstance. The growing complexity of real-world optimization problems has motivated researchers to search for a new method. The simulation of some natural phenomena, for example, physical, chemical, evolutionary issues and biological behavioral patterns, has been used to develop new nature-inspired algorithms, which have demonstrated flexibility, efficiency, simplicity and the avoidance of local optima more than traditional methods [51]. Inspired by this idea, ant colony optimization (ACO) [42,51], differential evolution (DE) [52], particle swarm optimization (PSO) [30], gray wolf optimization (GWO) [39,46], self-defense mechanism of plants optimization [7] have been proposed and applied widely. In recent years, these algorithms have been applied in many fields, such as image processing [4], power systems [5,34], data clustering [1], pattern recognition [21] and tuning of neural networks [40], bee colony optimization applied to fuzzy controller design [8] and the imperialist competitive algorithm applied to dynamic parameter optimization [3].

Moths often fly at night. Because they have a good sense of smell and hearing, they can adapt to life at night. Therefore, moths have phototaxis. From the onlooker's point of view, it is clear that moths are attracted by light, but

* Corresponding author at: College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China.
E-mail addresses: yongquanzhou@126.com, zhouyongquan@gxun.edu.cn (Y. Zhou).

https://doi.org/10.1016/j.matcom.2018.10.011
0378-4754/

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

the facts are more specific because moths become dizzy and confused around bright objects that are circling. Moths use light as a compass to navigate, which evolved into a fixed part of the eye that accepts light. Provided the light source is far away, such as the sun or moon, moths' eyes receive light at an angle that is almost identically parallel. Then, provided moths fly in a straight line toward the rectilinear direction, visual imaging remains unchanged. However, when the light source is very close, moths still fly straight, and the angle of the light that is received at each instant of movement changes. Therefore, to adapt to the change, moths, from the point of view of a bystander, seem to be spiraling toward the light source. Moths attempt to hide from predators in daylight, and at night they use the celestial navigation technique to orient themselves in the dark and exploit food sources. Moths fly in a straight line over a long distance by steering their locomotion at a constant angle relative to celestial far-distant point light, for example, moonlight, which is also known as the light compass reaction. Despite this, such orientation suffers from transverse direction motion because of futile spiral tracks around nearby artificial light sources [6,16,17,20].

The moth swarm algorithm (MSA) [41] was proposed by Al-Attar Ali Mohamed in 2016. It is a new population-based intelligent optimization algorithm inspired by the orientation of moths in a noisy environment toward moonlight. The MSA has good performance in the field of swarm intelligence, but its convergence precision and speed can be limited in some applications. In view of this deficiency, in this paper, the original algorithm is based on the elite opposition strategy to improve the convergence speed and accuracy.

The remainder of this paper is organized as follows: In Section 2, the original MSA is briefly introduced. This is followed in Section 3 by the new elite opposition-based MSA (EOMSA). Simulation experiments and results analysis are described in Section 4. Finally, the conclusion and future works are presented in Section 5.

2. Moth swarm algorithm (MSA)

In the MSA, a possible solution of the optimization problem is represented by the position of a light source, and the fitness of this solution is considered as the luminescence intensity of the light source. These assumptions are used to approximate the characteristics of the proposed algorithm. Additionally, the moth swarm is considered to consist of three groups of moths, which are defined as follows:

Pathfinders: a small group of moths that have the ability to discover new areas over the optimization space. The main task of this type is to discriminate the best positions of the light source to guide the movement of the main swarm.

Prospectors: a group of moths that tends to wander in a random spiral path within the neighborhood of light sources that have been marked by the pathfinders.

Onlookers: a group of moths that drift directly toward the best global solution, which has been obtained by prospectors.

For any iteration, each moth x_i is incorporated into the optimization problem to determine the luminescence intensity of its corresponding light source $f(x_i)$. The best fitness function in the swarm is considered as the positions of the pathfinders and guidance for the next update iteration. Hence, the second and third best groups take the names of prospectors and onlookers, respectively. The proposed optimization algorithm is executed in the following phases:

2.1. Pathfinder phase

At the beginning of the MSA, for a D-dimensional problem and n number of population, the positions of moths are randomly created as follows:

$$x_{ij} = rand\,[0, 1] \cdot \left(x_j^{\max} - x_j^{\min}\right) + x_j^{\min}, \tag{1}$$

where x_{ij} is the position of moth i in the jth dimension, and $x_j^{\max}$ and $x_j^{\min}$ are the upper and lower limits, respectively.

In some cases, the moths may sink into local optimization. To avoid premature convergence and increase the diversity of the population, a part of the swarm can jump out of the local optimal solution. The pathfinder moths update their position by interacting with crossover operations, and have the ability to fly for long distances using the proposed adaptive crossover with Lévy mutation, which is described as follows:

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

2.1.1. Choice of crossover points

A proposed strategy for the diversity is to select the crossover points. At iteration t, the normalized dispersal degree σ_j^t of the individuals in the jth dimension is measured as follows:

$$\sigma_j^t = \frac{\sqrt{\frac{1}{n_p}\sum_1^{n_p}\left(x_{ij}^t - \overline{x_j^t}\right)^2}}{\overline{x_j^t}}, \tag{2}$$

where n_p is the number of pathfinder moths. Then, variation coefficient μ^t, which is a measure of relative dispersion, is formulated as follows:

$$\mu^t = \frac{1}{d}\sum_{j=1}^{d}\sigma_j^t. \tag{3}$$

Crossover point group c_p is described as

$$j \in c_p \; if \;\; \sigma_j^t \;\; \leq \;\; \mu^t \;\; . \tag{4}$$

It can be seen that the group of crossover points changes dynamically with the progress of the strategy.

2.1.2. Lévy mutation

Lévy motions are random processes based on α-stable distribution with the ability to travel over large-scale distances using different sizes of steps. Lévy α-stable distribution is strongly linked to the heavy-tailed probability density function, fractal statistics and anomalous diffusion. Compared with Gaussian distribution and Cauchy distribution, Lévy distribution achieves a heavier tail. Mantegna's algorithm [15] is used to emulate α-stable distribution by generating random samples L_i that have the same behavior as Lévy flights as follows:

$$L_i \sim step \oplus Levy\,(\alpha) \sim 0.01\frac{\mu}{|y^{1/\alpha}|}, \tag{5}$$

where $step$ is the scale size related to the scale of the problem of interest, $\oplus$ denotes entrywise multiplication, and $\mu = N\left(0, \sigma_u^2\right)$ and $y = N\left(0, \sigma_y^2\right)$ are two normal stochastic distributions with $\sigma_\mu = \left[\frac{\Gamma(1+\alpha)\sin\pi\alpha/2}{\Gamma\left((1+\alpha)/2\alpha 2^{(\alpha-1)/2}\right)}\right]^{1/\alpha}$ and $\sigma_y = 1$.

2.1.3. Position update

For $n_c \in c_p$, the position update process is based on the differential evolution model. By disturbing the selected host vector $\overrightarrow{x_p} = \left[x_{p1}, x_{p2}, \ldots, x_{pn_c}\right]$, the algorithm creates sub-trial vector $\overrightarrow{v_p} = \left[v_{p1}, v_{p2}, \ldots, v_{pn_c}\right]$, with related vectors (e.g., $\overrightarrow{x_{r^1}} = \left[x_{r^1 1}, x_{r^1 2}, \ldots, x_{r^1 n_c}\right]$). The formula for the sub-trial vector is as follows:

$$\overrightarrow{v_p^t} = \overrightarrow{x_{r^1}^t} + L_{p1}^t \cdot \left(\overrightarrow{x_{r^2}^t} - \overrightarrow{x_{r^3}^t}\right) + L_{p2}^t \cdot \left(\overrightarrow{x_{r^4}^t} - \overrightarrow{x_{r^5}^t}\right) \tag{6}$$

$$\forall r^1 \neq r^2 \neq r^3 \neq r^4 \neq r^5 \neq p \in \left\{1, 2, \ldots, n_p\right\},$$

where L_{p1} and L_{p2} are two independent variables that are generated by Lévy α-stable distribution using the fifth expression. The set of indices is exclusively selected from the pathfinder solutions.

Each pathfinder solution (host vector) updates its position by incorporating the mutated variables of the sub-trail vector into the corresponding variables of the host vector. The completed trail solution v_{pj} is described as

$$v_{pj}^t = \begin{cases} v_{pj}^t & if \quad j \in c_p \\ x_{pj}^t & if \quad j \notin c_p \end{cases} \tag{7}$$

After the previous procedure is complete and compared with the host solution, the fitness value of the sub-trail solution is calculated. The fitter solutions are selected for the next generation, which is outlined for the minimization problem as follows:

$$\overrightarrow{x_p^{t+1}} = \begin{cases} \overrightarrow{x_p^t} & if \quad f(\overrightarrow{v_p^t}) \geq f(\overrightarrow{x_p^t}) \\ \overrightarrow{v_p^t} & if \quad f(\overrightarrow{v_p^t}) < f(\overrightarrow{x_p^t}) \end{cases} \tag{8}$$

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

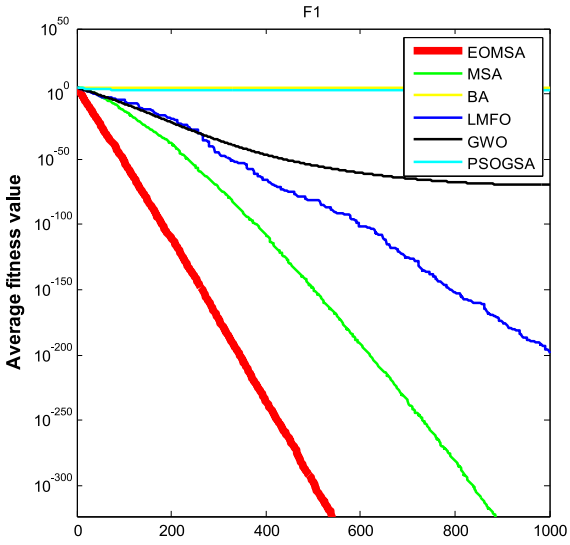

Fig. 1. $D = 30$: evolution curves of the fitness values for f_1.

Probability value p_e is estimated to be proportional to the luminescence intensity fit_p as follows:

$$p_e = \frac{fit_p}{\sum_{p=1}^{n_p} fit_p}. \tag{9}$$

Luminescence intensity fit_p is calculated from objective function value f_p for minimization problems as

$$fit_p = \begin{cases} \frac{1}{1+f_p} & for \quad f_p \geq 0 \\ 1 + |f_p| & for \quad f_p < 0 \end{cases} \tag{10}$$

2.2. Prospector phase

Because of transverse orientation, the prospector moths fly in a logarithmic spiral path. The prospector moths are considered to be the next best luminescence intensity group. The number of prospectors n_e is processed to decrease with the course of iterations T as follows:

$$n_e = round\left((n - n_c) \times \left(1 - \frac{t}{T}\right)\right). \tag{11}$$

After the pathfinders complete their search, the luminescence intensities are transferred to the prospectors, which update their positions to discover new light sources. Each prospector x_i makes a deep search around corresponding artificial light source x_e in a logarithmic spiral manner. Artificial light source x_e (according to the pathfinder moths' found positions) is selected depending on probability p_e using Eq. (9). The new position of the ith prospector moth is expressed as follows:

$$x_i^{t+1} = |x_i^t - x_p^t| \cdot e^{\theta} \cdot \cos 2\pi\theta + x_p^t \tag{12}$$

$$\forall p \in \{1, 2, \ldots, n_p\}; i \in \{n_p + 1, n_p + 2, \ldots, n_f\},$$

where $\theta \in [r, 1]$ is a random number that defines the spiral shape and $r = -1 - t/T$. The same expression is used in the moth-flame optimization (MFO) [37] algorithm, but each variable is processed as an integrated unit in the MSA. In the MSA, the type of each moth is dynamically selected. Hence, when a prospector moth finds a solution with more luminescence than the existing light sources, it becomes a pathfinder moth. Of course, the new light sources merged at the end of this stage.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

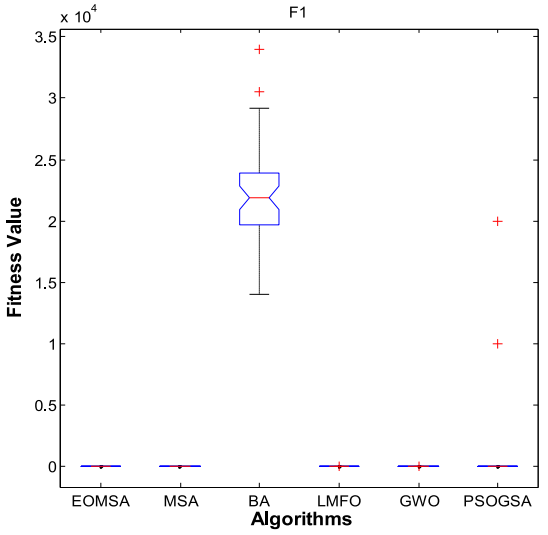

Fig. 2. $D = 30$: ANOVA test of the global minimum for f_1.

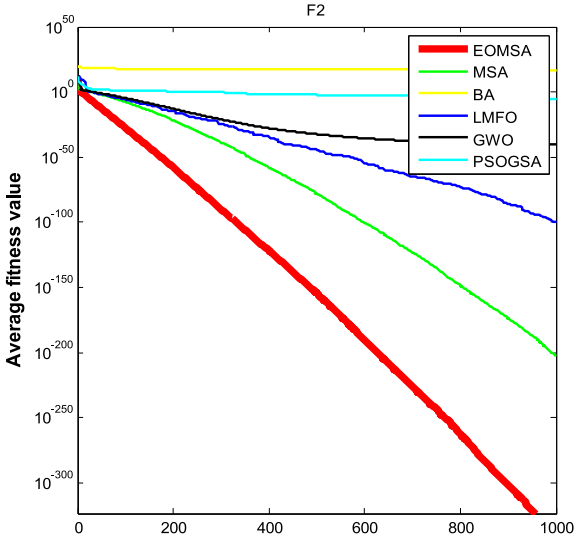

Fig. 3. $D = 30$: evolution curves of the fitness values for f_2.

2.3. Onlooker phase

The moths that have the lowest luminescent intensities in the swarm are considered onlookers. In the optimization process, the decrease in the number of prospectors results in an increase in the number of onlookers ($n_u = n - n_e - n_c$), which leads to a rapid increase in the convergence speed of the mentioned algorithm toward a global solution. These moths fly directly toward the area where the light is most intense (the moon). In this phase, the MSA is designed to force the onlookers to search more effectively by zooming in on the hot spots of the prospectors. The onlookers are divided into the following two subgroups.

2.3.1. Gaussian walks

In the first part, the number of onlookers equals $n_o = round(n_u/2)$. The new onlookers walk according to Gaussian distributions using Eq. (13). These moths in subgroup x_i^{t+1} move with series steps of Gaussian walks using Eq. (14), which is described as follows:

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

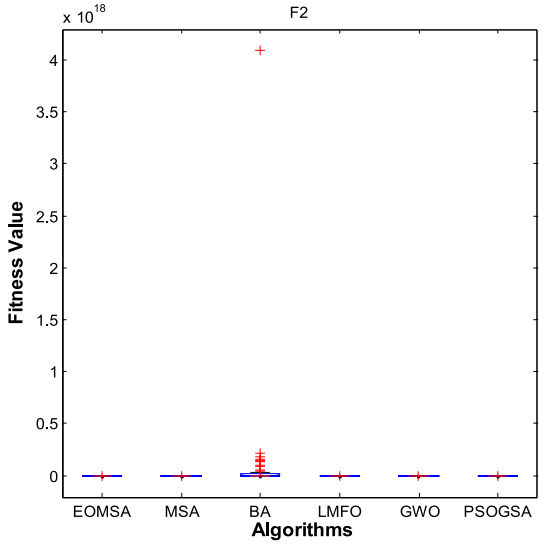

Fig. 4. $D = 30$: ANOVA test of the global minimum for f_2.

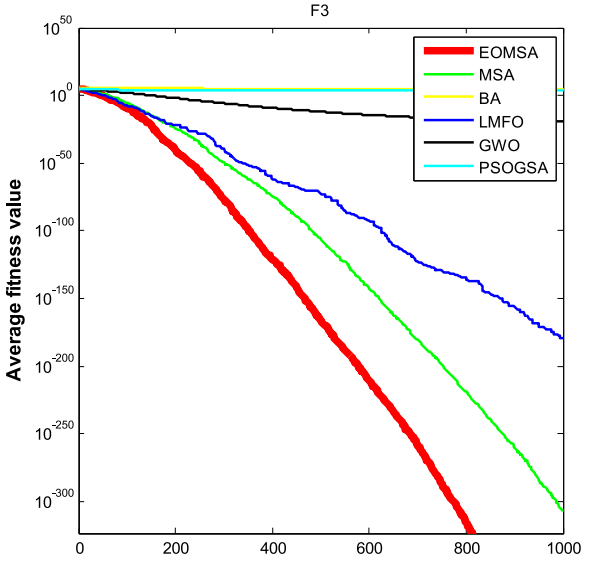

Fig. 5. $D = 30$: evolution curves of the fitness values for f_3.

Gaussian distribution $q \sim N(\mu, \sigma_G^2)$ if the density is

$$f(q) = \frac{1}{\sqrt{2\pi}\sigma_G} \exp\left(-\frac{(q-\mu)^2}{2\sigma_G^2}\right) \quad -\infty < q < \infty \quad q \sim N(\mu,) \tag{13}$$

$$x_i^{t+1} = x_i^t + \varepsilon_1 + \left[\varepsilon_2 \times gbest^t - \varepsilon_3 \times x_i^t\right] \quad \forall i \in \{1, 2, \ldots, n_o\} \tag{14}$$

$$\varepsilon_1 \sim random(size(d)) \oplus N(best_g^t, \frac{\log t}{t}(x_i^t - best_g^t)), \tag{15}$$

where ε_1 is a random sample drawn from Gaussian distributions scaled to the size of this group, $best_g$ is the global best solution obtained in the transverse orientation phase (both prospectors and pathfinders), and ε_2 and ε_3 are random numbers distributed uniformly within the interval [0, 1].

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

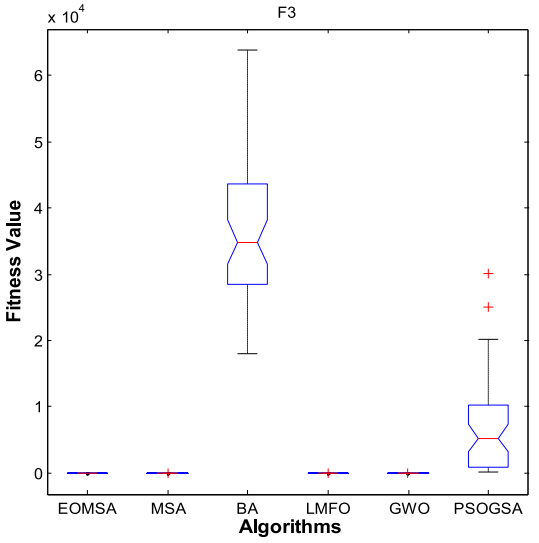

Fig. 6. $D = 30$: ANOVA test of the global minimum for f_3.

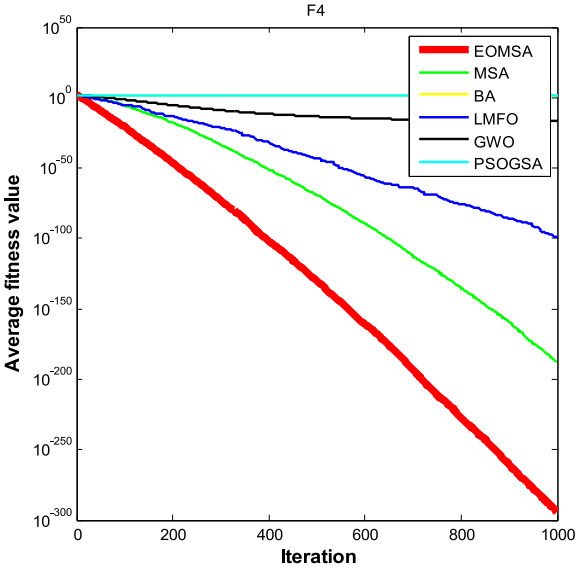

Fig. 7. $D = 30$: evolution curves of the fitness values for f_4.

2.3.2. *Associative learning mechanism with immediate memory*

In the second part, the number of onlookers equals $n_m = n_u - n_o$. In many evolutionary algorithms, memory exists to convey information from one generation to the next generation. However, in the real world, moths fall into fire because they do not have an evolutionary memory. The behavior of moths is strongly influenced by associative learning and short-term memory (1–3 s) [11,36]. Associative learning plays an important role in the communication between moths [14,50]. The second part of onlookers are designed to drift toward moonlight according to associative learning operators and have an immediate memory to mimic the actual behavior of moths in nature. The immediate memory is initialized from the continuous uniform Gaussian distribution on the interval from $x_i^t - x_i^{\min}$ to $x_i^{\max} - x_i^t$. The updating equation for this type is

$$x_i^{t+1} = x_i^t + 0.001 \cdot G\left[x_i^t - x_i^{\min}, x_i^{\max} - x_i^t\right] + (1 - g/G) \cdot r_1(best_p^t - x_i^t) + 2g/G \cdot r_2 \cdot (best_g^t - x_i^t), \quad (16)$$

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

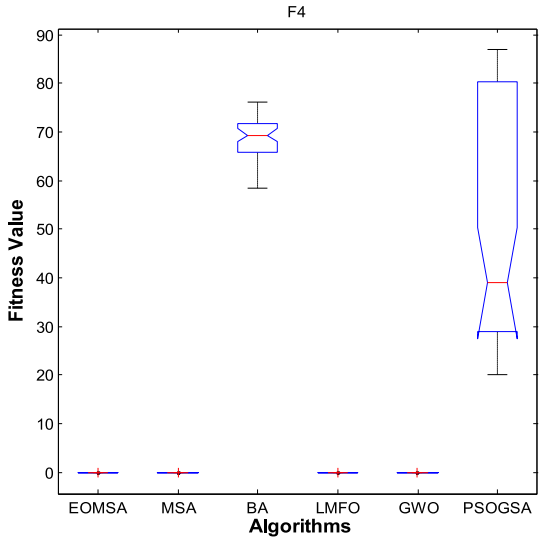

Fig. 8. $D = 30$: ANOVA test of the global minimum for f_4.

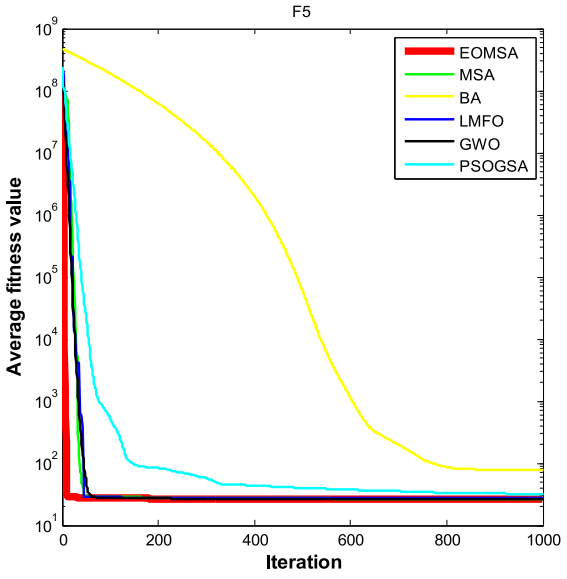

Fig. 9. $D = 30$: evolution curves of the fitness values for f_5.

where $i \in \{1, 2, \ldots, n_m\}$, $2g/G$ is the social factor, $1 - g/G$ is the cognitive factor, and r_1 and r_2 are random numbers within the interval [0, 1]. Similar to Section 1, $best_p$ is a light source randomly chosen from the new pathfinder group based on the probability value of its corresponding solution.

At the end of each iteration, the fitness value of the entire population is used to redefine the type of each moth for the next iteration. The specific implementation steps of the standard MSA are summarized in the pseudocode shown in Algorithm 1.

3. Elite opposition-based moth swarm algorithm (EOMSA)

The MSA (Al-Attar Ali Mohamed, 2016) can easily solve low-dimensional unimodal optimization problems and high-dimensional multimodal optimization problems. To speed up the convergence speed and convergence accuracy of the MSA, we add an elite strategy to improve the performance of the algorithm.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

Algorithm 1 Moth swarm algorithm (MSA)

Step 1: Initialize the moth swarm population using Eq. (1).

Step 2: Calculate the swarm fitness, sort all the moth individuals as per their fitness and identify the type of each moth.

Step 3: While (t < Maxg) **do**

For each pathfinder moth

Define the crossover points using Eqs. (2), (3) and (4)

Generate Lévy flight samples L_p using Eq. (5)

Create sub-trail vector $\overrightarrow{v_p^t}$ using Eq. (6)

Construct complete trail solution v_p^t using Eq. (7)

Select the artificial light sources $\overrightarrow{x_p^{t+1}}$ using Eq. (8)

Calculate probability values p using Eqs. (9) and (10)

End for

For each prospector moth x_i

Update the position of the prospector moth using Eq. (12)

Calculate the fitness of the prospector moth

Define the new light sources and moonlight

End for

For each onlooker moth

Update the position according to its type

If ($i \in n_0$)

Generate Gaussian walk steps $\varepsilon_1, \varepsilon_2, \varepsilon_3$ using Eq. (13)

Move the onlooker position with Gaussian walk x_i^{t+1} using Eq. (14)

Else

Drift the onlooker moth using the associative learning operators and immediate memory using Eq. (15)

End if

Calculate the fitness of the onlooker moth

End for

Identify the new light sources, moonlight and type of each moth

End while

Print the global best solution (moonlight).

The added elite reverse mechanism increases the exchange of information between moth individuals and elite moth individuals. To a certain extent, the elite reverse strategy enlarges the search space of the algorithm, enhances the diversity of the population and enhances the ability of the algorithm to explore the local optimal solution.

3.1. Elite opposition-based learning strategy (EOLS)

In the initial stage of the moth algorithm, the type of light source and moth is defined according to the size of the fitness value. The initialization is a random process and the fitness values are random. In the initial stage, adding the elite reverse learning mechanism to regenerate the moth's position and calculate the fitness value can make the algorithm better toward the better direction.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

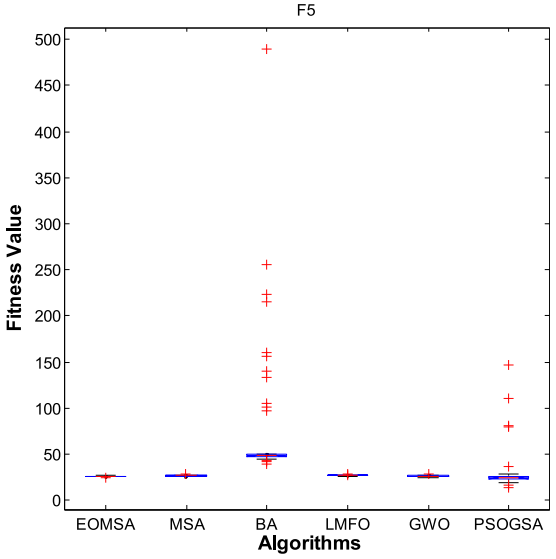

Fig. 10. $D = 30$: ANOVA test of the global minimum for f_5.

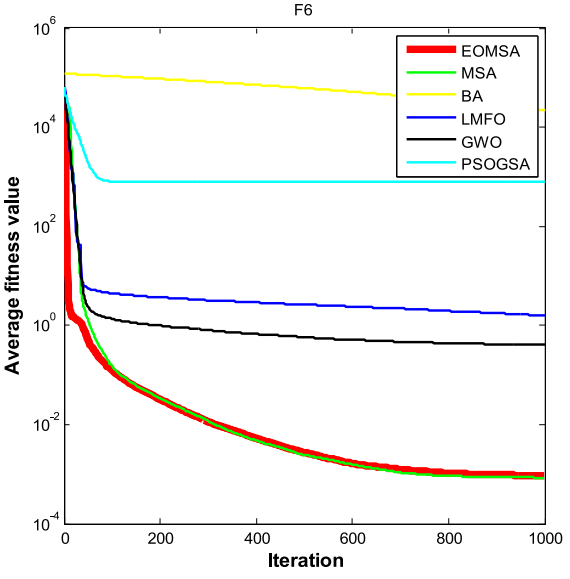

Fig. 11. $D = 30$: evolution curves of the fitness values for f_6.

Elite opposition-based learning is a new technique in the field of intelligence computation. Its main ideology is as follows: For a feasible solution, calculate and evaluate the opposite solution at the same time and choose the better solution as the individual for the next generation. In this paper, the individual with the best fitness value in the population is viewed as an elite individual. An example is provided to explain the definition of the elite opposition-based solution. Suppose the elite individual of the population is $x_e = (x_{e1}, x_{e2}, \ldots, x_{eD})$. For individual $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, the elite opposition-based solution of x_i is defined as x_i'. Additionally, it can be obtained by

$$x'_{i,j} = k \cdot \left(da_j + db_j\right) - x_{e,j} \tag{17}$$
$$i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, D$$

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

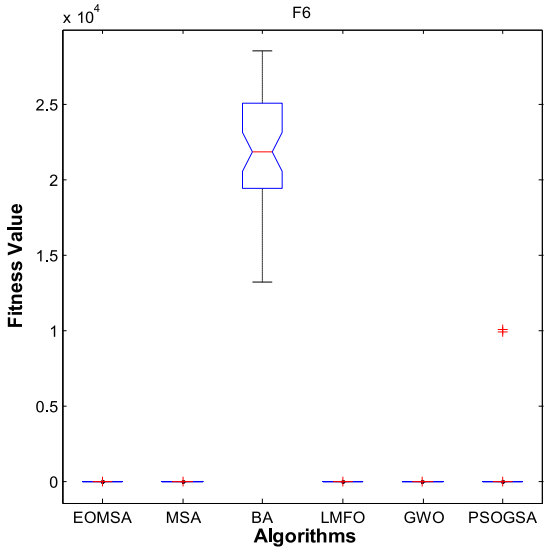

Fig. 12. $D = 30$: ANOVA test of the global minimum for f_6.

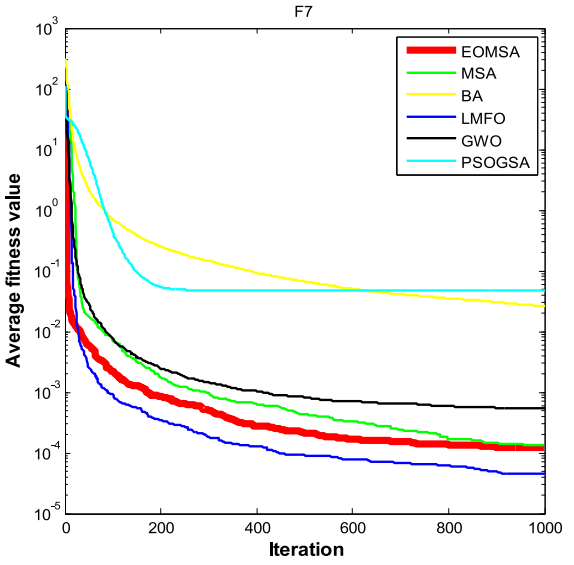

Fig. 13. $D = 30$: evolution curves of the fitness values for f_7.

where n is the population size, D is the dimension of x, $k \in U\ (0, 1)$, and da_j and db_j are the dynamic bounds of the jth decision variable. da_j and db_j can be obtained by

$$da_j = \min(x_{i,j}) \quad db_j = \max\left(x_{i,j}\right). \tag{18}$$

In the global search process, this strategy expands the search space of the algorithm, and it can strengthen the diversity of the population; thus, the proposed global search ability can be enhanced using this optimization strategy. EOMSA can be summarized in the pseudocode shown in Algorithm 2.

4. Simulation experiments and result analysis

In this section, 23 benchmark functions [22,53] are applied to evaluate the optimal performance of the EOMSA. These 23 benchmark functions (shown in Table 1) have been widely used in the literature. The EOMSA is compared

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

Algorithm 2 Elite opposition-based moth swarm algorithm (EOMSA)

Step 1: Initialize the moth swarm population using Eq. (1)

Use EOLS to generate the elite opposition-based solution $x_i^{'}$ using Eq. (17)

Step 2: Calculate the swarm fitness, sort all the moth individuals as per their fitness and identify the type of each moth

Step 3: While （t < Maxg） **do**

For each pathfinder moth

Define the crossover points using Eqs. (2), (3) and (4)

Generate Lévy flight samples L_p using Eq. (5)

Create sub-trail vector $\overrightarrow{v_p^t}$ using Eq. (6)

Construct complete trail solution v_p^t using Eq. (7)

Select artificial light sources $\overrightarrow{x_p^{t+1}}$ using Eq. (8)

Calculate probability values p using Eqs. (9) and (10)

End for

For each prospector moth x_i

Update the position of the prospector moth using Eq. (12)

Calculate the fitness of the prospector moth

Define the new light sources and moonlight

End for

For each onlooker moth

Update the position according to its type

If ($i \in n_0$)

Generate Gaussian walk steps $\varepsilon_1, \varepsilon_2, \varepsilon_3$ using Eq. (13)

Move the onlooker position with Gaussian walk x_i^{t+1} using Eq. (14)

Else

Drift the onlooker moth using the associative learning operators and immediate memory using

Eq. (15)

End if

Calculate the fitness of the onlooker moth

End for

Identify the new light sources, moonlight and type of each moth

End while

Print the global best solution (moonlight).

with four state-of-the-art metaheuristic algorithms: bat algorithm (BA) [55], LMFO [32], GWO [39], PSO-GSA [38] and MSA [41]. The space dimension, scope, optimal value and iterations of the 23 functions are shown in Table 1. The remainder of this section is organized as follows: The experimental setup is provided in Section 4.1, the comparison of each algorithm's performance is provided in Section 4.2 and in Section 4.3, three structural design examples (welded beam design [12], compression spring design [31] and pressure vessel design optimization [9]) are used to evaluate the EOMSA's ability to solve a constrained function optimization problem.

The selected benchmark functions can be divided into three categories: high-dimensional unimodal functions (category I), high-dimensional multimodal functions (category II) and fixed-dimension multimodal functions (category III). They are f_1, f_2, f_3, f_4, f_5, f_6 and f_7 for category I; f_8, f_9, f_{10}, f_{11}, f_{12}, and f_{13} for category II; and

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

Table 1
Twenty-three benchmark functions.

Category	No	Name	Benchmark functions	D	Scope	$f_{\min}$
I	F1	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	30	$x_i \in [-100, 100]$	0
	F2	Schwefel2.22	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$x_i \in [-10, 10]$	0
	F3	Schwefel1.2	$f(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$x_i \in [-100, 100]$	0
	F4	Schwefel2.21	$f(x) = max_i \lvert x_i \rvert, 1 \le i \le D$	30	$x_i \in [-100, 100]$	0
	F5	Resonbrock	$f(x) = \sum_{i=1}^{D-1} [100 \left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2]$	30	$x_i \in [-30, 30]$	0
	F6	Step	$f(x) = \sum_{i=1}^{n} \left(\lvert x_i + 0.5 \rvert\right)^2$	30	$x_i \in [-100, 100]$	0
	F7	Quartic	$f(x) = \sum_{i=1}^{n} x_i^4 + random(0, 1)$	30	$x_i = [-1.28, 1.28]$	0
II	F8	Generalized	$f(x) = -\sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	$x_i \in [-500, 500]$	−418.9829n
	F9	Rastrigin	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$x_i \in [-5.12, 5.12]$	0
	F10	Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp(\frac{1}{n}\sum_{i=1}^{n} cos2\pi x_i)\right) + 20 + e$	30	$x_i \in [-32, 32]$	0
	F11	Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} \left(x_i^2\right) - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$x_i \in [-600, 600]$	0
	F12	Penalty#1	$f_{12}(x) = 0.1\left\{\sin^2\left(3\pi x_1 + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right)\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]	0
	F13	Penalty#2	$f(x) = 0.1\left\{\sin^2\left(3\pi x_1 + \sum_{i=1}^{n} (x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right)\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$x_i \in [-50, 50]$	0

(*continued on next page*)

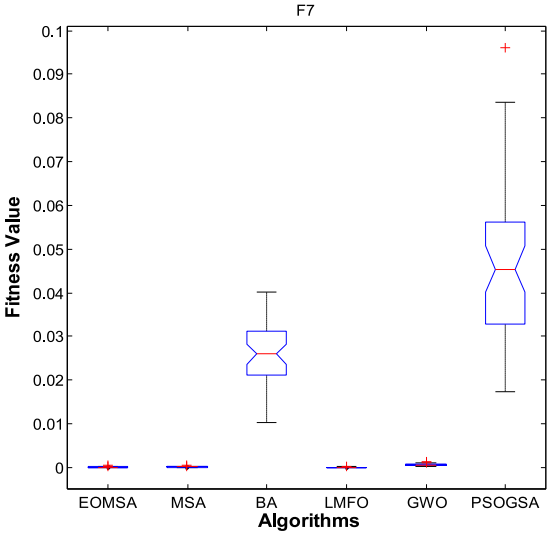

Fig. 14. $D = 30$: ANOVA test of the global minimum for f_7.

Table 1 (*continued*).

Category	No	Name	Benchmark functions	D	Scope	$f_{\min}$
III	F14	Shekel's Foxholes	$f(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	$x_i \in [-65, 65]$	1
	F15	Kowalik	$f(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	$x_i \in [-5, 5]$	0.0003075
	F16	Drop wave	$f(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5\left(x_1^2 + x_2^2\right) + 2}$	2	$x_i \in [-5.12, 5.12]$	−1
	F17	Six hump camel back	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$x_i \in [-5, 5]$	−1.0316285
	F18	GoldStein Price	$f(x) = \left[1 + (x_1 + x_2 + 1)^2\left(\begin{array}{c} 19 - 14x_1 + 3x_1^2 - 14x_2 + \\ 6x_1x_2 + 3x_2^2 \end{array}\right)\right]$ $\times[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$x_i \in [-5, 5]$	3
	F19	Hartman	$f(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	$x_i \in [1, 3]$	−3.86
	F20	Hartman	$f(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	$x_i \in [0, 1]$	−3.32
	F21	Shekel 1	$f(x) = -\sum_{i=1}^{5}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$x_i \in [0, 10]$	−10.1532
	F22	Shekel 2	$f(x) = -\sum_{i=1}^{7}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$x_i \in [0, 10]$	−10.4029
	F23	Shekel 3	$f(x) = -\sum_{i=1}^{10}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$x_i \in [0, 10]$	−10.5364

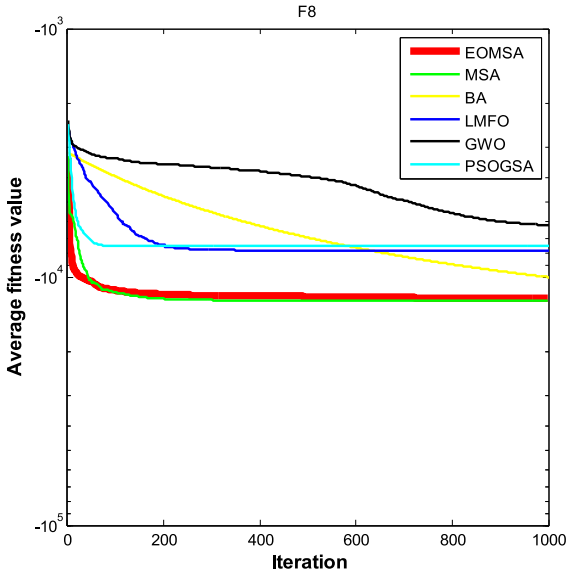

Fig. 15. $D = 30$: evolution curves of the fitness values for f_8.

f_{14}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}, f_{21}, f_{22}, and f_{23} for category III. High-dimensional unimodal functions have one global optimum and no local optima. However, multimodal functions have a tremendous number of local optima and are helpful to examine the exploration and local optima avoidance of algorithms. In the benchmark functions, f_5 is a classical test function. Its global minimum is in a parabolic valley, and the function values change little in the valley. Thus, it is very difficult to find the global minimum. There are a large number of local minima in the solution space of f_{10}.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

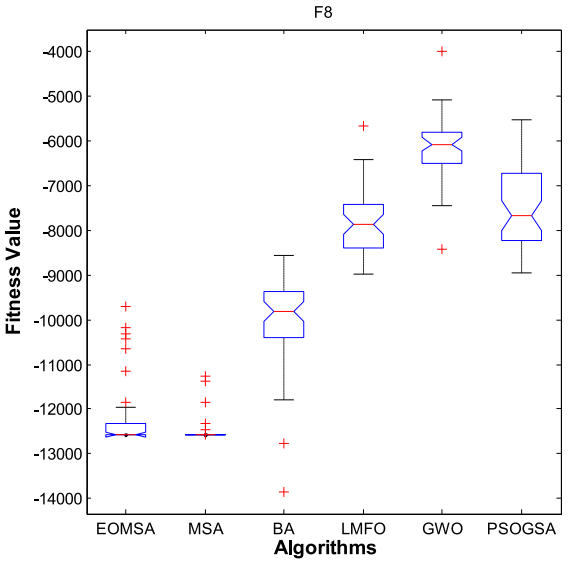

Fig. 16. $D = 30$: ANOVA test of the global minimum for f_8.

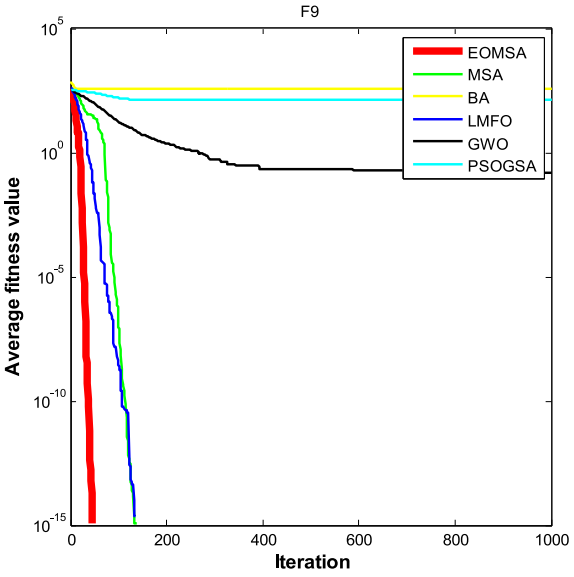

Fig. 17. D=30: evolution curves of the fitness values for f_9.

4.1. Experimental setup

The development environment for this test was MATLAB R2012a. The test ran on an AMD Athlon (tm) II*4640 processor and 4 GB memory.

4.2. Comparison of each algorithm's performance

The proposed EOMSA is compared with swarm intelligence algorithms BA [55], LMFO [32], GWO [39], PSO-GSA [38] and MSA [41] using the mean and standard deviation to compare the optimal performance. The settings for the algorithm control parameters are given as follows:

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

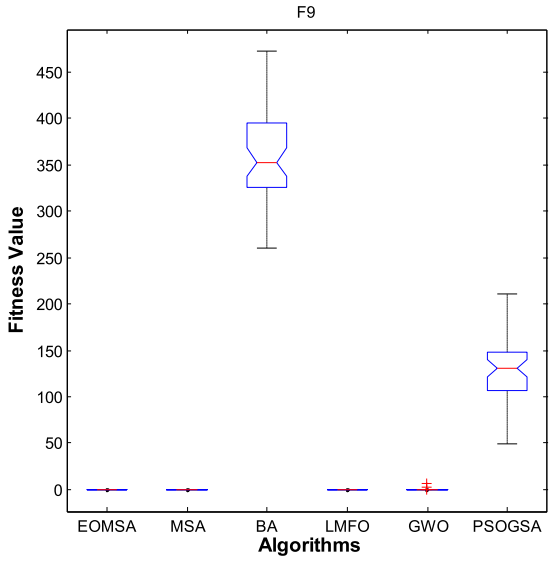

Fig. 18. D=30: ANOVA test of the global minimum for f_9.

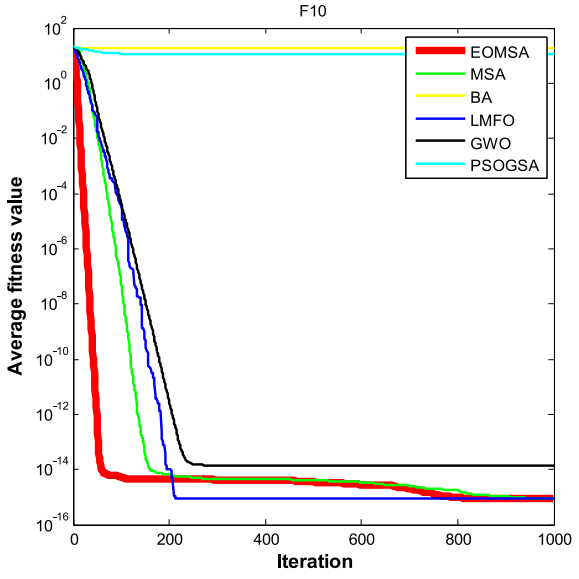

Fig. 19. D = 30: evolution curves of the fitness values for f_{10}.

BA settings: maximum pulse intensity A is 0.9, maximum pulse frequency r is 0.5, pulse attenuation coefficient α is 0.95, pulse frequency increase factor γ is 0.05 [55], population size is 50 and maximum iteration number is 1000.

MFO settings: population size is 50 and maximum iteration number is 1000.

GWO settings: $\vec{\alpha}$ linearly decreasing from 2 to 0 is used as recommended in [39], population size is 50 and maximum iteration number is 1000.

PSO-GSA settings: population size is 50 and maximum iteration number is 1000.

MSA settings: population size is 50 with six pathfinders and maximum iteration number is 1000.

EOMSA setting: population size is 50 with six pathfinders and maximum iteration number is 1000.

To maintain consistency for the comparison, these algorithms were tested for 50 independent runs under the same conditions and using their standard control parameter settings as given. Fifty independent tests of the three categories of standard benchmark functions (unimodal benchmark functions, multimodal benchmark functions and

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

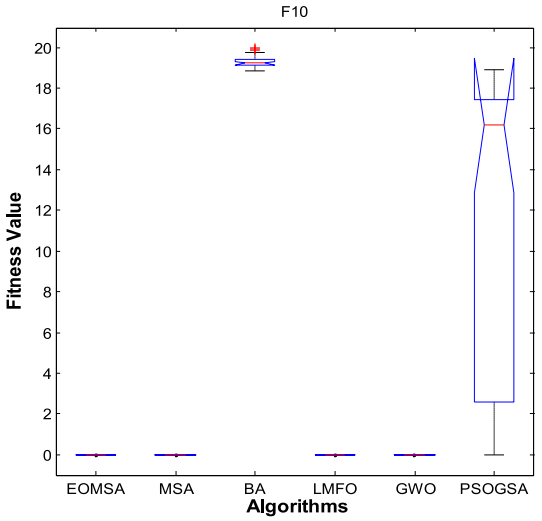

Fig. 20. $D = 30$: ANOVA test of the global minimum for f_{10}.

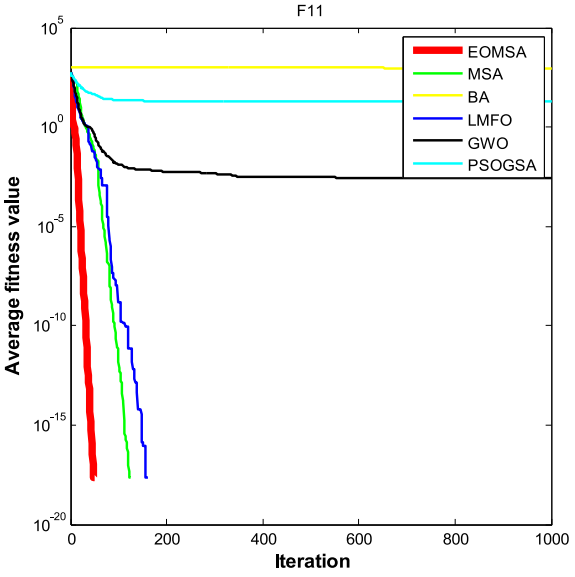

Fig. 21. $D = 30$: evolution curves of the fitness values for f_{11}.

fixed-dimension multimodal benchmark functions) given in Table 1 were conducted. Their results are shown in Tables 2–4, respectively. In the tables, the best fitness value, worst fitness value, average fitness value and standard deviation are represented by Best, Worst, Mean and Std, respectively. The mean and standard deviation were used to assess the robustness of the algorithm. All algorithms were ranked according to the value of Std.

According to the experimental results in Table 2, the EOMSA reached the optimal value from f_1 to f_4. The EOMSA did not reach the optimal value from f_5 to f_6, but its convergence rate and convergence accuracy in the comparison of the six algorithms were the highest. For the seventh benchmark function, the EOMSA also did not reach the optimal solution, and it was not closer to the optimal solution than the LMFO algorithm. In the early stage, the EOMSA converged faster than other algorithms. From the mean and standard deviation, it is easy to see that the EOMSA had higher robustness. From F_1 to F_7, the EOMSA was more stable; there was no shock. From Figs. 1–14, we can more intuitively see that the performance of the EOMSA was superior to other algorithms. From the convergence diagram,

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

Table 2
Simulation results for the test functions.

Benchmark functions	Result	Method						Rank
		PSO-GSA	LMFO	BA	GWO	MSA	EOMSA	
f_1 ($D = 30$)	Best	1.50E−19	1.80E−260	14541.83	9.35E−62	0	0	1
	Worst	20000	1.50E−197	29285.14	1.72E−57	0	0	
	Mean	2000	3.00E−199	346017.73	1.37E−58	0	0	
	Std	4517.54	0	3324.663	3.36E−58	0	0	
f_2 ($D = 30$)	Best	1.56E−09	9.90E−132	1.29E+10	9.48E−36	5.50E−215	0	1
	Worst	9.49E−05	1.60E−98	4.09E+18	4.50E−34	1.50E−199	0	
	Mean	1.90E−06	3.20E−100	1.07E+17	9.60E−35	3.20E−201	0	
	Std	1.34E−05	2.30E−99	5.77E+17	9.72E−35	0	0	
f_3 ($D = 30$)	Best	116.1154	8.80E−239	18044.45	1.82E−20	0	0	1
	Worst	30047.04	2.00E−178	63807.42	6.13E−15	4.80E−306	0	
	Mean	6800.916	3.90E−18	36835.74	6.15E−16	9.70E−308	0	
	Std	7129.106	0	11107.11	1.44E−15	0	0	
f_4 ($D = 30$)	Best	20.20454	5.00E−124	58.57462	7.79E−16	1.50E−201	0	1
	Worst	86.93711	2.95E−98	76.06284	1.63E−13	1.40E−187	1.10E−296	
	Mean	52.16009	7.80E−100	68.71811	2.08E−14	2.90E−189	2.20E−298	
	Std	26.00567	4.30E−99	4.425038	3.34E−14	0	0	
f_5 ($D = 30$)	Best	14.19328	26.42829	39.57799	25.78455	25.56954	24.60994	2
	Worst	146.6106	28.77259	489.1546	27.93699	28.71312	26.86511	
	Mean	32.09675	27.02958	79.4571	26.77688	26.77739	25.91146	
	Std	24.45746	0.393261	78.07746	0.550832	0.703707	0.486618	
f_6 ($D = 30$)	Best	1.38E−19	1.041491	13252.65	1.34E−05	0.000131	5.61E−05	1
	Worst	10100.25	2.020635	28555.37	1.499963	0.003391	0.002985	
	Mean	800.02	1.558201	22072.17	0.604857	0.000852	0.000646	
	Std	2740.693	0.220464	3668.607	0.351231	0.000581	0.000556	
f_7 ($D = 30$)	Best	0.01738	1.47E−06	0.010311	9.21E−05	4.10E−06	9.80E−06	2
	Worst	0.096062	0.000197	0.040134	0.002009	0.000537	0.00042	
	Mean	0.047261	4.44E−05	0.026092	0.000929	0.000132	0.000124	
	Std	0.018094	4.45E−05	0.00719	0.000449	0.000108	0.000102	

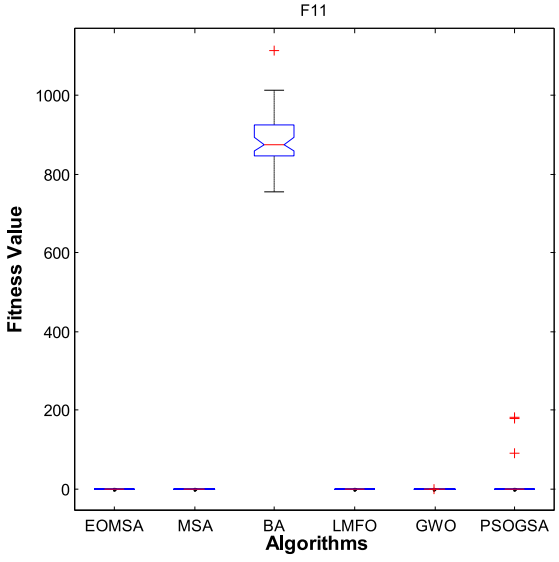

Fig. 22. $D = 30$: ANOVA test of the global minimum for f_{11}.

Table 3
Simulation results for the test functions.

Benchmark functions	Result	Method						Rank
		PSO-GSA	LMFO	BA	GWO	MSA	EOMSA	
f_8 ($D = 30$)	Best	−8941.94	−8992.38	−13847.6	−7134.58	−12569.5	−12569.5	1
	Worst	−5516.5	−5671.52	−8565.2	−2978.59	−8516.28	−9231.97	
	Mean	−7502.99	−7867.74	−10012.6	−5865.43	−12492.9	−12376.6	
	Std	870.6463	701.7211	982.6808	1051.649	573.3431	561.7044	
f_9 ($D = 30$)	Best	48.75286	0	260.391	0	0	0	1
	Worst	210.93	0	472.3588	3.284771	0	0	
	Mean	130.3418	0	360.1401	0.218035	0	0	
	Std	34.49052	0	45.82105	0.829769	0	0	
f_{10} ($D = 30$)	Best	2.93E−10	8.88E−16	18.84569	1.15E−14	8.88E−16	8.88E−16	1
	Worst	18.92663	8.88E−16	19.96441	2.22E−14	8.88E−16	8.88E−16	
	Mean	11.31789	8.88E−16	19.33333	1.62E−14	8.88E−16	8.88E−16	
	Std	7.262999	0	0.328353	2.66E−15	0	0	
f_{11} ($D = 30$)	Best	0	0	753.7714	0	0	0	1
	Worst	180.4568	0	1113.01	0.038614	0	0	
	Mean	19.84469	0	884.3628	0.005362	0	0	
	Std	45.65812	0	67.28712	0.009894	0	0	
f_{12} ($D = 30$)	Best	0.000539	0.064187	25.15514	0.008198	5.76E−06	4.02E−06	1
	Worst	11.37178	0.159308	58.416	0.081365	0.103757	5.91E−05	
	Mean	3.314715	0.108771	37.29893	0.035399	0.002106	2.21E−05	
	Std	3.181618	0.024343	8.06297	0.019022	0.014669	1.28E−05	
f_{13} ($D = 30$)	Best	1.57E−05	1.190871	142.9433	0.215339	0.000133	7.44E−05	1
	Worst	38.38423	2.385186	198.5082	0.84965	0.386286	0.002286	
	Mean	11.2787	1.694802	175.0042	0.516178	0.046499	0.000517	
	Std	9.704188	0.258309	11.48439	0.198198	0.080511	0.000382	

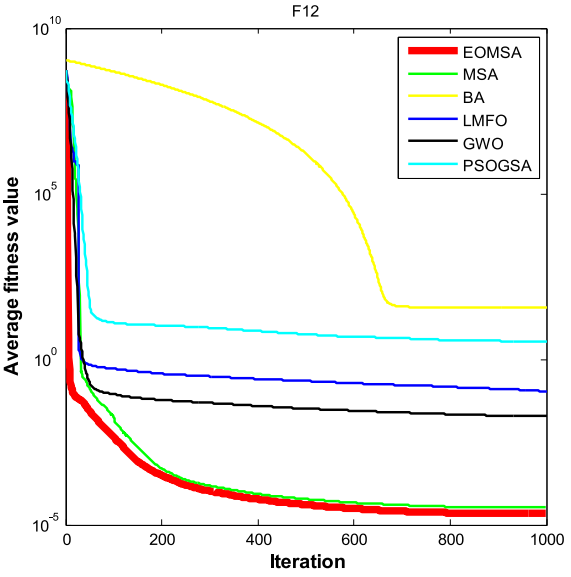

Fig. 23. $D = 30$: evolution curves of the fitness values for f_{12}.

we can see that the convergence speed and accuracy of the EOMSA were relatively fast; only for the f_7 convergence was accuracy slightly worse than that for LMFO.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

Table 4
Simulation results for the test functions.

Benchmark functions	Result	Method						Rank
		PSO-GSA	LMFO	BA	GWO	MSA	EOMSA	
f_{14} $(D = 2)$	Best	0.998004	0.998004	1.992031	0.998004	0.998004	0.998004	1
	Worst	21.98841	12.67051	23.80943	12.67051	10.76318	10.76318	
	Mean	1.892663	2.84577	12.1125	5.42555	2.175802	1.388611	
	Std	3.131832	3.085824	6.400806	4.901748	2.700248	1.933004	
f_{15} $(D = 4)$	Best	0.000307	0.000311	0.000308	0.000307	0.000307	0.0003075	2
	Worst	0.020363	0.001228	0.004857	0.001594	0.020363	0.001224	
	Mean	0.003701	0.00035	0.001006	0.000447	0.001739	0.001204	
	Std	0.007356	0.000129	0.000687	0.000353	0.004769	0.000251	
f_{16} $(D = 2)$	Best	−1	−1	−0.9362	−1	−1	−1	1
	Worst	−1	−1	−0.4778	−1	−1	−1	
	Mean	−1	−1	−0.7914	−1	−1	−1	
	Std	0	0	0.14225	0	0	0	
f_{17} $(D = 2)$	Best	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	1
	Worst	−1.03163	−1.0316	−0.21546	−1.03163	−1.03163	−1.03163	
	Mean	−1.03163	−1.03162	−0.8956	−1.03163	−1.03163	−1.03163	
	Std	6.05E−16	6.9E−06	0.309366	4.59E−09	6.39E−16	2.99E−16	
f_{18} $(D = 2)$	Best	3	3	3	3	3	3	3
	Worst	84	3.000089	30	3.000049	30	30	
	Mean	4.62	3.000023	4.620001	3.000011	4.62	4.62	
	Std	11.45513	2.47E−05	6.477244	1.4E−05	6.477244	6.477224	
f_{19} $(D = 3)$	Best	−3.86278	−3.86251	−3.86278	−3.86278	−3.86278	−3.86278	1
	Worst	−3.86278	−3.85333	−3.08976	−3.8549	−3.86278	−3.86278	
	Mean	−3.86278	−3.86101	−3.84732	−3.86193	−3.86278	−3.86278	
	Std	2.98E−15	0.001554	0.109321	0.002188	2.91E−15	2.78E−15	
f_{20} $(D = 6)$	Best	−3.322	−3.30742	−3.32191	−3.32199	−3.322	−3.322	3
	Worst	−3.2031	−3.1324	−3.20285	−3.08668	−3.2031	−3.2031	
	Mean	−3.32023	−3.21844	−3.26002	−3.25563	−3.27026	−3.28871	
	Std	0.000956	0.051188	0.059971	0.082353	0.059276	0.053925	
f_{21} $(D = 4)$	Best	−10.1532	−5.04446	−10.1532	−10.153	−10.1532	−10.1532	1
	Worst	−2.63047	−4.9546	−2.63046	−5.05518	−10.1532	−10.1532	
	Mean	−5.28663	−5.01092	−6.19176	−9.64446	−10.1532	−10.1532	
	Std	3.085186	0.023745	3.393087	1.550791	2.72E−05	4.11E−08	
f_{22} $(D = 4)$	Best	−10.4029	−6.4248	−10.4029	−10.4029	−10.4029	−10.4029	4
	Worst	−1.83759	−4.98424	−1.83759	−5.08766	−3.7243	−2.7659	
	Mean	−6.29321	−5.09427	−5.56415	−10.2253	−10.2694	−9.84948	
	Std	3.466064	0.238418	3.337365	0.970348	1.322027	1.420189	
f_{23} $(D = 4)$	Best	−10.5364	−5.86911	−10.5364	−10.5363	−10.5364	−10.5364	3
	Worst	−1.67655	−5.00082	−1.85948	−5.17515	−2.80663	−3.83543	
	Mean	−6.10111	−5.0978	−5.08178	−10.3573	−9.97975	−10.2684	
	Std	3.872679	0.11437	3.512583	0.978752	1.911115	1.32645	

The experimental results from Table 3 show that only PSO-GSA and BA did not find the optimal solution for f_9. Only BA did not find the optimal solution for f_{11}. The EOMSA found the optimal solution better than other algorithms (solving the minimum value) for f_8, f_{10}, f_{12} and f_{13}. It is obvious that the EOMSA clearly escaped from the poor local optimum, and the EOMSA and MSA approached the neighborhood of the global optimum for f_{12}; whereas the BA had a low probability regarding making such a long jump, which may be the reason for its poor average best fitness. From the function convergence graph in Figs. 15–26, we can clearly observe that the EOMSA's convergence rate and convergence accuracy were better than other algorithms under the same conditions. From the variogram, we can clearly observe that the EOMSA was more stable, except for f_8 under the same conditions, which indicates better robustness.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

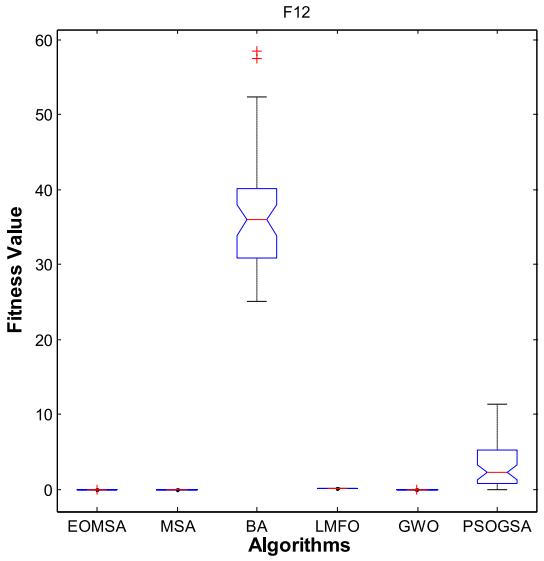

Fig. 24. $D = 30$: ANOVA test of the global minimum for f_{12}.

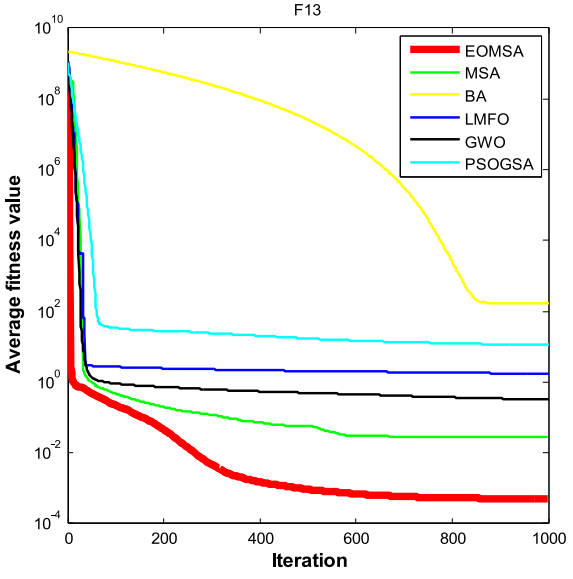

Fig. 25. $D = 30$: evolution curves of the fitness values for f_{13}.

From the function convergence graph in Figs. 27–46, we can see the f_{17} optimum value was -1; there was no logarithm of the y-axis. We can see from Table 4 that, although f_{14} was an easy problem, the EOMSA failed to find the optimal value. However, the EOMSA found the optimal value for f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}, f_{21}, f_{22}, f_{23}. Because it is a low-dimensional multimodal function, the comparison of the five algorithms can find some optimal value, and some are caught in the local optimal solution. The mean value and std indicate the robustness of the algorithm. The mean value of the EOMSA was higher than that of LMFO for f_{15}, so the robustness of LMFO was better than that of the EOMSA. Similar to the EOMSA, the MSA had the same ability to find the optimal value, but the convergence rate of the MSA was lower than that of the EOMSA. From the convergence diagram Figs. 27–46, to summarize, the algorithms achieved a similar performance ranking for both multimodal categories, where the EOMSA was ranked Located in the top three.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

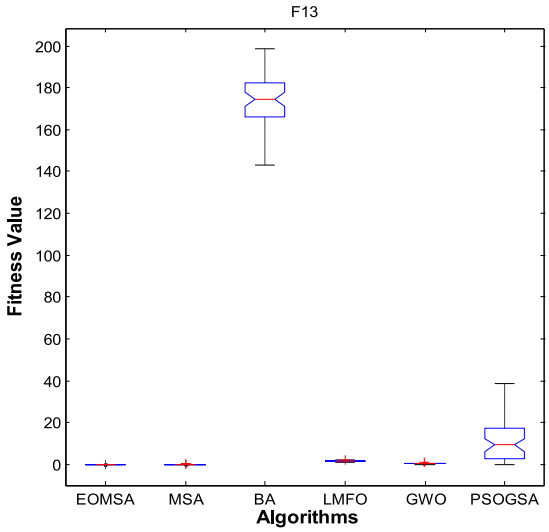

Fig. 26. $D = 30$: ANOVA test of the global minimum for f_{13}.

Table 5
p-values of the Wilcoxon rank-sum test results ($p \geq 0.05$ has been underlined).

Functions	MSA	BA	GWO	LMFO	PSO-GSA
f_{01}	N/A	3.31E−20	3.31E−20	3.31E−20	3.3E−20
f_{02}	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20
f_{03}	0.001823	3.31E−20	3.31E−20	3.31E−20	3.31E−20
f_{04}	7.07E−18	7.07E−18	7.07E−18	7.07E−18	7.07E−18
f_{05}	4.67E−08	7.07E−18	6.28E−05	2.03E−14	0.000838
f_{06}	0.034605	7.07E−18	4.73E−09	7.07E−18	4.65E−13
f_{07}	0.009257	7.07E−18	4.01E−16	5.27E−05	7.07E−18
f_{08}	2.73E−05	2.05E−13	7.07E−18	7.07E−18	7.07E−18
f_{09}	N/A	3.31E−20	0.000943	N/A	3.31E−20
f_{10}	N/A	3.31E−20	7.79E−21	N/A	3.31E−20
f_{11}	N/A	3.31E−20	0.003462	N/A	1.69E−18
f_{12}	0.025737	7.07E−18	4.65E−13	7.07E−18	7.07E−18
f_{13}	1.17E−09	7.07E−18	8.99E−18	7.07E−18	1.7E−16
f_{14}	0.790116	1.41E−13	1E-09	8.41E−10	0.000612
f_{15}	0.010644	2.56E−11	0.70713	4.5E−10	0.001238
f_{16}	0.567589	4.73E−20	4.73E−20	4.73E−20	7.69E−15
f_{17}	0.754483	2.14E−19	2.14E−19	2.14E−19	0.012277
f_{18}	0.121991	7.62E−12	1.84E−11	1.84E−11	0.00064
f_{19}	0.774853	2.57E−18	2.57E−18	2.57E−18	0.038987
f_{20}	0.790589	1.74E−08	3.55E−08	8.66E−10	0.001821
f_{21}	0.110624	2.68E−18	2.68E−18	2.68E−18	6.82E−09
f_{22}	0.325766	5.23E−16	2.8E−14	2.65E−14	0.009422
f_{23}	0.67336	3.73E−17	1.2E−15	1.5E−15	0.000853

4.3. p-values of the Wilcoxon's rank-sum test

To validate the comparative study, the pairwise Wilcoxon's rank-sum test [18,54], a nonparametric statistical test, was conducted at a 0.05 significance level to assess whether the results of the EOMSA differed from the other five algorithms in a statistical manner. The p-values of the Wilcoxon's rank-sum are presented in Table 5. In this table, the p-values less than 0.05 provide sufficient evidence against the null hypothesis.

Hypothesis testing is an important part of inferential statistics. The p-value is commonly found in hypothesis testing. It is another basis for the testing decision. The p-value is the probability, and reflects the likelihood of an event

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

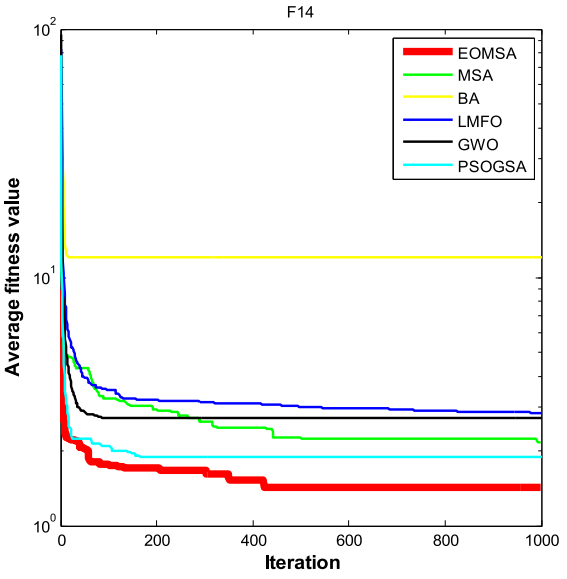

Fig. 27. $D = 30$: evolution curves of the fitness values for f_{14}.

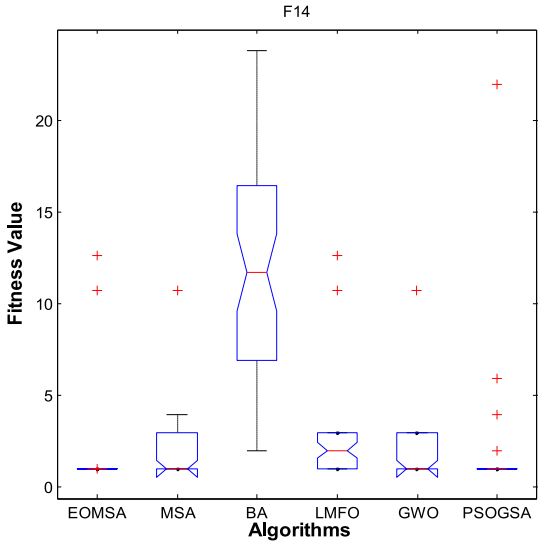

Fig. 28. $D = 30$: ANOVA test of the global minimum for f_{14}.

size. The p-value obtained by the statistical test according to the significance test is generally significant at $p < 0.05$; $p < 0.01$ is very significant, which means that the difference between the samples is less than 0.05 or 0.01 because of the sampling error.

When $p > 0.05$, this means that the chance of occurrence is more than 5%, which indicates no significant difference between the two groups. $p < 0.05$ indicates that the chance of occurrence is less than 5%, which indicates that the difference between the two groups is significant. $p < 0.01$ means that the chance of occurrence is less than 5%, which indicates that the difference between the two groups is very significant. The p-values were not significantly different between the EOMSA and MSA for f_{14}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}, f_{21}, f_{22}, f_{23}. They were similar between EOMSA and MSA for f_{15}.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

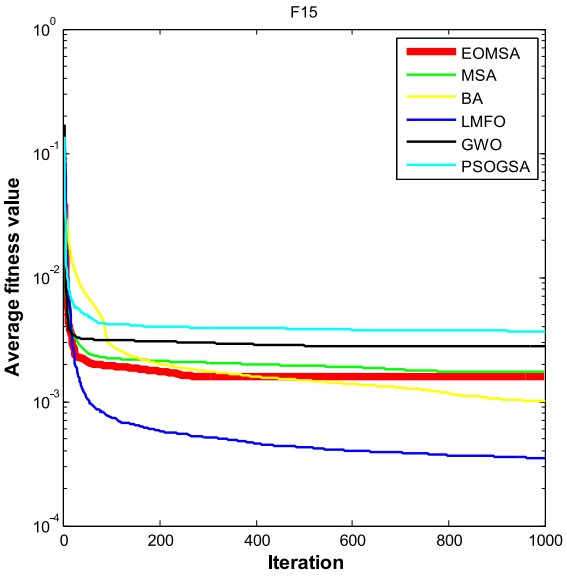

Fig. 29. $D = 30$: evolution curves of the fitness values for f_{15}.

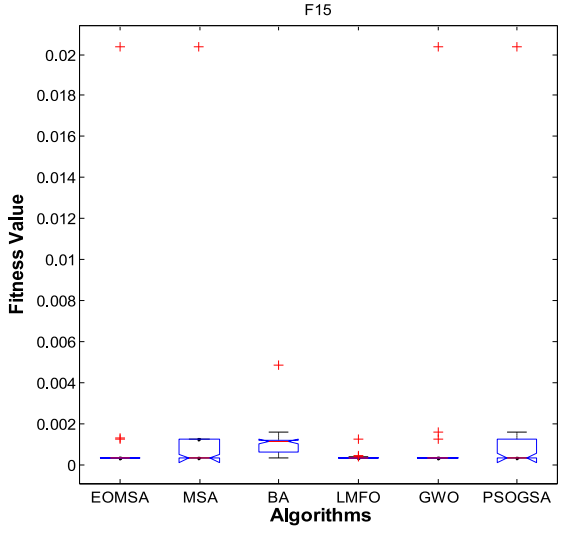

Fig. 30. $D = 30$: ANOVA test of the global minimum for f_{15}.

4.4. EOMSA for an engineering optimization problem

In this section, the EOMSA is tested on three constrained real-world engineering design problems: the welded beam, tension/compression spring and pressure vessel. Different types of penalty functions can be used with evolutionary algorithms to solve constraint problems [9]. Comparing the EOMSA and other evolutionary algorithms, we can assess the performance of the EOMSA in solving engineering problems.

4.4.1. Welded beam design optimization

Welded beam design (see Fig. 47) is considered as a standard constrained engineering problem. The objective function of this test problem aims to minimize the fabrication cost of the welded beam [48]. The problem involves four optimization variables: length of the clamped bar (l), thickness of the weld (h), thickness of the bar (b) and height

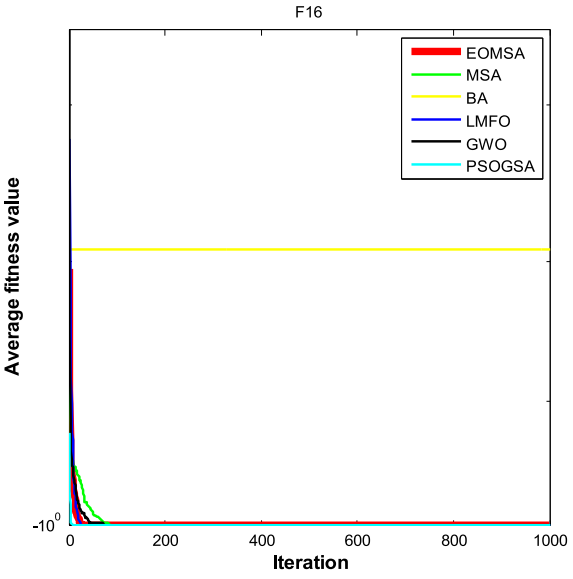

Fig. 31. $D = 30$: evolution curves of the fitness values for f_{16}.

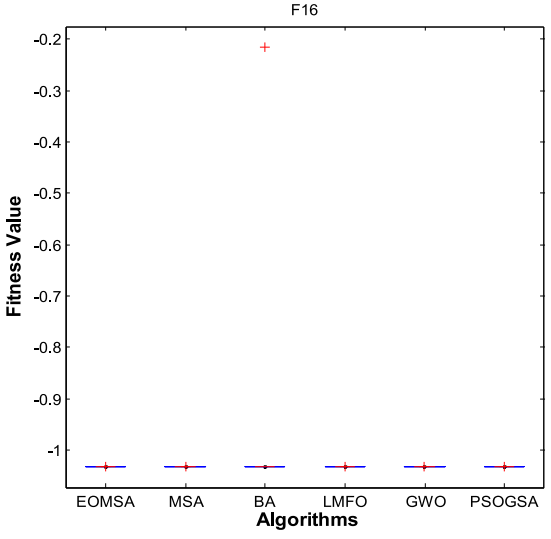

Fig. 32. $D = 30$: ANOVA test of the global minimum for f_{16}.

of the bar (t). The optimization constraints are bending stress in the beam (r), shear stress (s), end deflection of the beam (d) and buckling load on the bar (p_c). The formula involved in the design of a welded beam is as follows:
Consider $\overrightarrow{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [h \quad l \quad t \quad b \quad @]$.
Minimize $f(\overrightarrow{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$
subject to $g_1(\overrightarrow{x}) = \tau(\overrightarrow{x}) - \tau_{\max} \leq 0$,

$$g_2(\overrightarrow{x}) = \sigma(\overrightarrow{x}) - \sigma_{\max} \leq 0,$$

$$g_3(\overrightarrow{x}) = \delta(\overrightarrow{x}) - \delta_{\max} \leq 0,$$

$$g_4(\overrightarrow{x}) = x_1 - x_4 \leq 0,$$

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

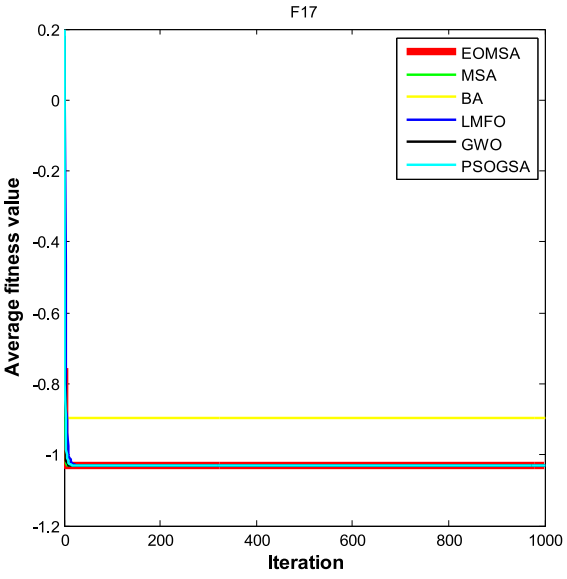

Fig. 33. $D = 30$: evolution curves of the fitness values for f_{17}.

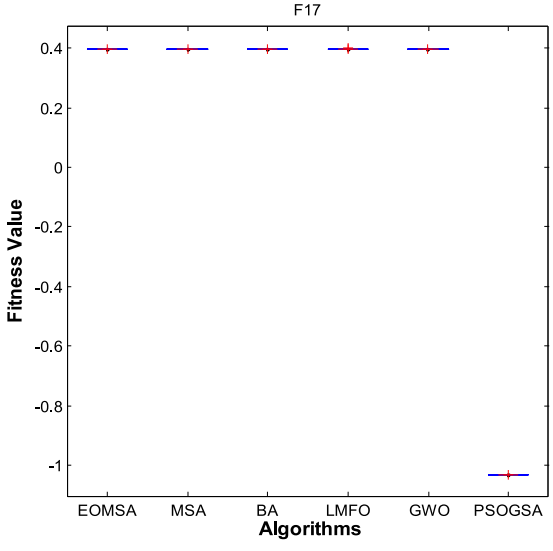

Fig. 34. $D = 30$ ANOVA test of the global minimum for f_{17}.

$$g_5(\vec{x}) = p - p_c(\vec{x}) \le 0,$$

$$g_6(\vec{x}) = 0.125 - x_1 \le 0,$$

$$g_7(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0, \tag{19}$$

where $\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$, $\tau' = \frac{p}{\sqrt{2}x_1x_2}$, $\tau'' = \frac{MR}{J}$, $M = p(L + \frac{x_2}{2})$,

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}, \quad J = 2\sqrt{2}x_1x_2R, \quad \sigma(\vec{x}) = \frac{6PL}{x_3^2x_4}, \quad \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4},$$

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

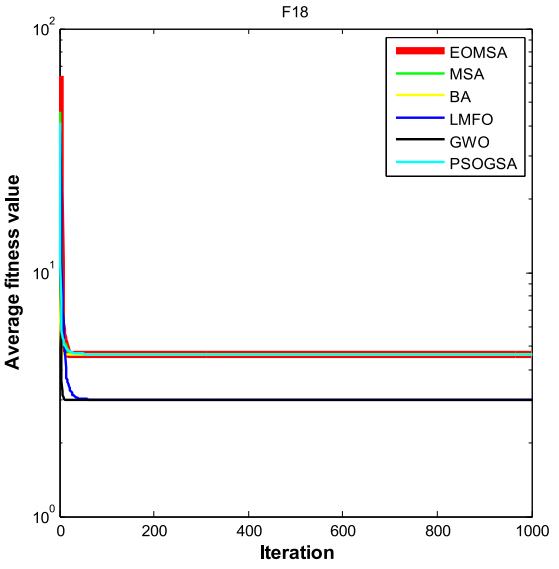

Fig. 35. $D = 30$: evolution curves of the fitness values for f_{18}.

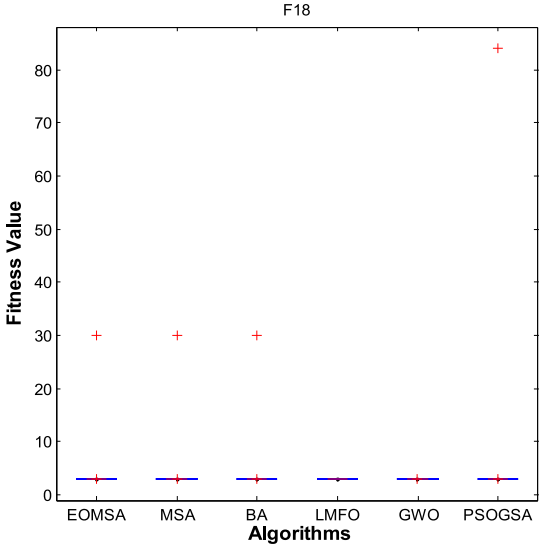

Fig. 36. $D = 30$: ANOVA test of the global minimum for f_{18}.

$$p_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$
$$p = 6000lb, \quad L = 14in., \quad \delta_{\max} = 0.25in., \quad E = 30 \times 10^6 psi, \quad G = 12 \times 10^6 psi,$$
$$\tau_{\max} = 13600psi \quad \sigma_{\max} = 30,000psi.$$

with boundary conditions $0.1 \leq x_1, x_4 \leq 2, \quad 0.1 \leq x_2, x_3 \leq 10$.

The test was conducted independently 25 times. The test results are shown in Table 6. The results for the EOMSA and GWO [39], BA [55], co-evolutionary DE (CDE) [25], water cycle algorithm (WCA) [13], mine blast algorithm (MBA) [47], improved ACO (IACO) [29], ray optimization (RO) [28], society and civilization (SaC) algorithm [45], hybrid PSO and DE (PSO-DE) [33], and weighted superposition attraction (WSA) [2] for the 25 independent experiments to verify the validity of EOMSA for the welding beam problem are shown in Table 6.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

Table 6
Comparison results for the welded beam design problem.

Algorithm	Optimal values for variables				Optimal cost
	h	l	t	b	
EOMSA	0.22425	3.2486	8.6518	0.22445	1.7246
GWO [39]	0.205676	3.478377	9.03681	0.205778	1.72624
BA [55]	0.2015	3.562	9.0414	0.2057	1.7312065
CDE [25]	0.20317	3.542998	9.033498	0.206179	1.733462
WCA [13]	0.205728	3.470522	9.036620	0.205729	1.724856
MBA [47]	0.205729	3.470493	9.036626	0.205729	1.724853
IACO [29]	0.205700	3.471131	9.036683	0.205731	1.724918
RO [28]	0.203687	3.528467	9.004263	0.207241	1.735344
SaC [45]	0.244438	6.237967	8.288576	0.244566	2.3854347
PSO-DE [33]	N/A	N/A	N/A	N/A	1.7248531
WSA [2]	0.2057296	3.4704899	9.0366239	0.2057296	1.72485254

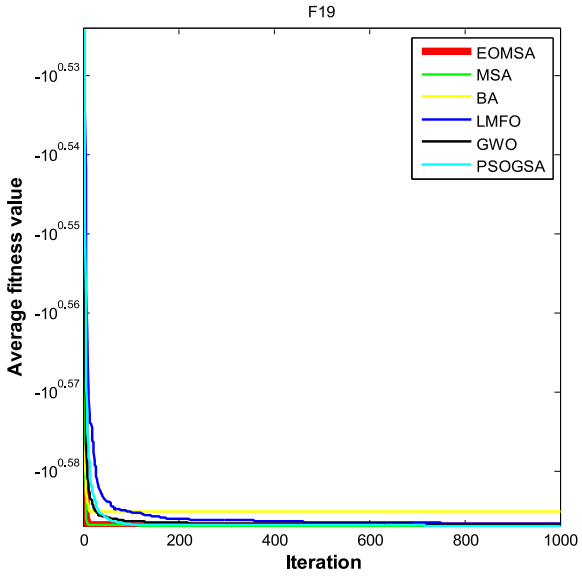

Fig. 37. $D = 30$: evolution curves of the fitness values for f_{19}.

It can be seen from Table 6 that the EOMSA found an optimal value for the welding beam problem that was smaller than that found for the other algorithms, which shows that the EOMSA performed better than the other algorithms for the welding beam problem, which would save costs in a practical application.

4.4.2. Tension/compression spring design optimization

The spring design optimization problem (see Fig. 48) is used to test the superiority of an optimization method. In this problem, the goal is to minimize the weight of the tension/compression spring. The optimum design must satisfy constraints on surge frequency, shear stress and deflection. The problem involves three design variables: mean coil diameter (D), wire diameter (d) and number of active coils (N). The formula involved in the design of tension/compression spring design optimization is as follows:

Consider $\overrightarrow{x} = [x_1 \quad x_2 \quad x_3] = [d \quad D \quad N]$.

Minimize $f(\overrightarrow{x}) = (x_3 + 2)x_2x_1^2$,

subject to $g_1(\overrightarrow{x}) = 1 - \frac{x_3x_2^3}{71785x_1^4} \leq 0$,

$$g_2(\overrightarrow{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0,$$

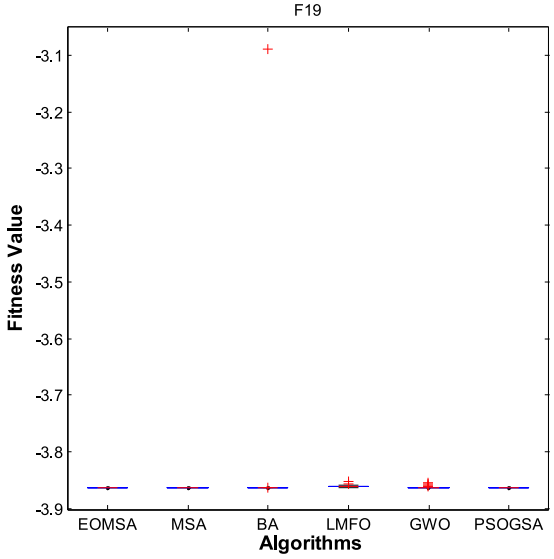

Fig. 38. $D = 30$: ANOVA test of the global minimum for f_{19}.

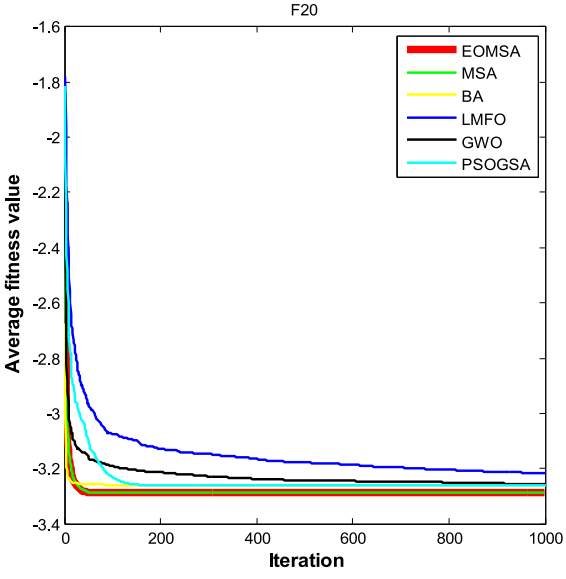

Fig. 39. $D = 30$: evolution curves of the fitness values for f_{20}.

$$g_3(\vec{x}) = 1 - \frac{140x_1}{x_3 x_2^2} \leq 0,$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0, \tag{20}$$

and variable range $0.05 \leq x_1 \leq 2.0, \quad 0.25 \leq x_2 \leq 1.3, \quad 2.0 \leq x_3 \leq 15.0$.

Several optimization algorithms were applied to this problem petal44645: RO [28], SaC [45], bacterial gene recombination algorithm (BGRA) [24], improved harmony search algorithm (IHS) [35], hybrid Nelder–Mead simplex search and PSO (NM-PSO) [56], co-evolutionary PSO (CPSO) [23] and yin–yang pair optimization (YYPO) [43]. Table 7 shows the comparison of the statistical results obtained by the EOMSA and those algorithms.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

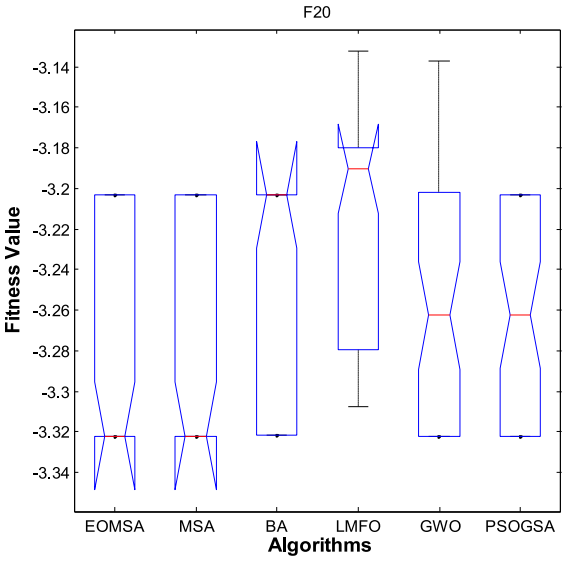

Fig. 40. $D = 30$: ANOVA test of the global minimum for f_{20}.

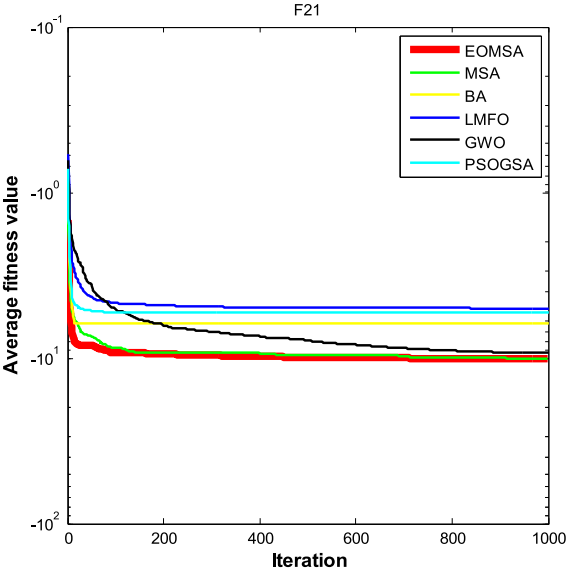

Fig. 41. $D = 30$: evolution curves of the fitness values for f_{21}.

From the comparison of several algorithms in Table 7, it can be seen that the EOMSA obtained an optimal value smaller than that obtained by the other algorithms. This indicates that the EOMSA obtained a better value for this problem. This would reduce costs and achieve greater efficiency in practical problems.

4.4.3. Pressure vessel design optimization

The pressure vessel design (see Fig. 49) is a classic hybrid constrained optimization problem [18], and includes the cost of material, and forming and welding of a cylindrical pressure vessel capped at both ends by hemispherical heads. The problem involves four optimization constraints and four design variables: thickness of the head (T_h), thickness of the shell (T_s), length of the cylindrical section without considering the head (L) and inner radius (R). The formula involved in the design of pressure vessel design optimization is as follows:

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

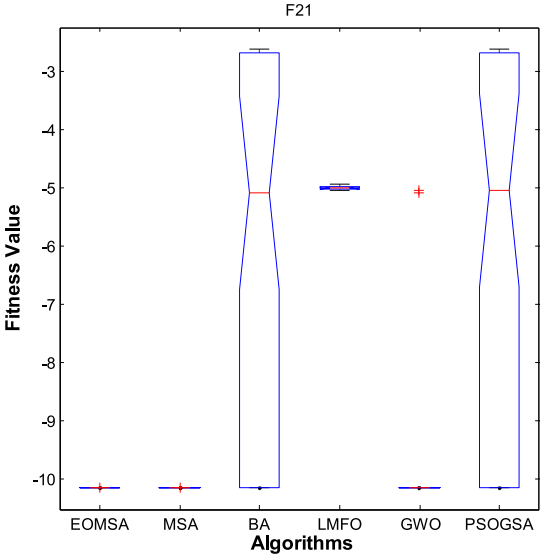

Fig. 42. $D = 30$: ANOVA test of the global minimum for f_{21}.

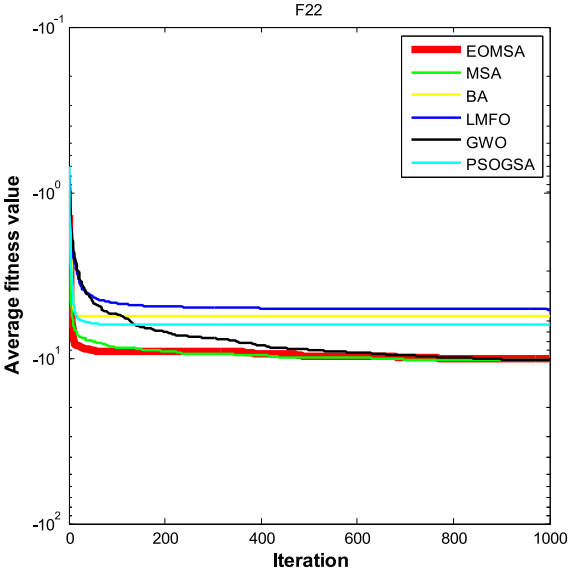

Fig. 43. $D = 30$: evolution curves of the fitness values for f_{22}.

Consider $\overrightarrow{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [T_s \quad T_h \quad R \quad L]$.
Minimize $f(\overrightarrow{x}) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$
subject to $g_1(\overrightarrow{x}) = -x_1 - 0.0193x_3 \leq 0$,
$g_2(\overrightarrow{x}) = -x_3 - 0.00954x_3 \leq 0$,
$g_3(\overrightarrow{x}) = -\pi x_4x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$,
$g_4(\overrightarrow{x}) = x_4 - 240 \leq 0$,
and variable range $0.0 \leq x_1, x_2 \leq 99.0, \quad 10.0 \leq x_3, \quad x_4 \leq 200.0$

Heuristic methods (GSA [44], GA [10], WEO [27] and PSO-GA [19]) and mathematical methods (Lagrangian multiplier [26] and branch-and-bound [49]) were adopted to optimize this problem. Table 8 shows that the EOMSA found the minimum cost for all algorithms.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

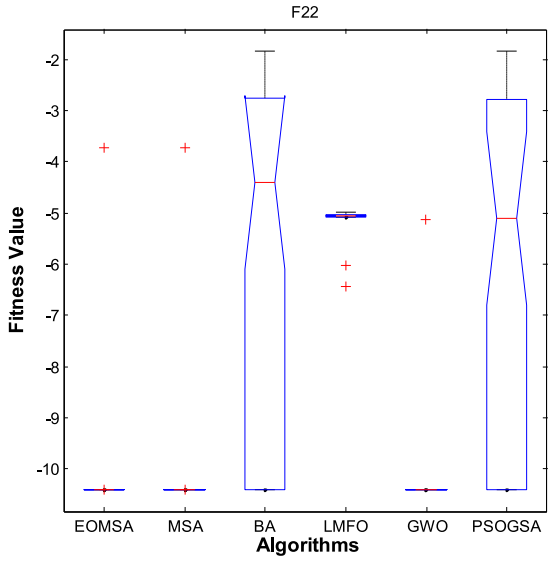

Fig. 44. $D = 30$: ANOVA test of the global minimum for f_{22}.

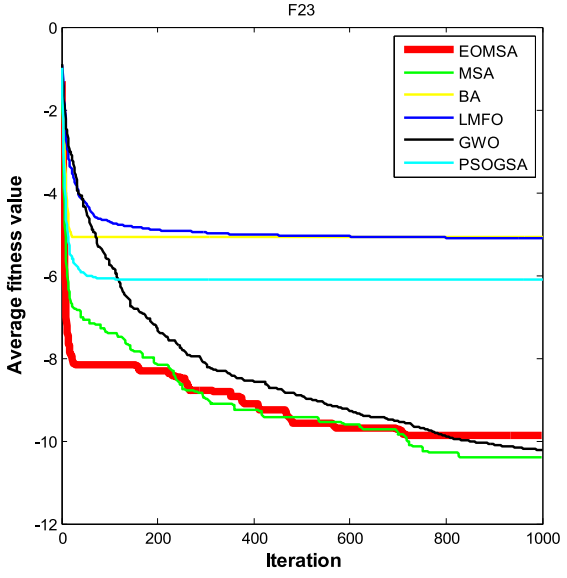

Fig. 45. $D = 30$: evolution curves of the fitness values for f_{23}.

5. Conclusions

In this paper, to overcome the disadvantage of standard MSA, an optimization strategy was incorporated into the MSA to generate the EOMSA for function optimization and structure engineering design problems. Global elite opposition-based learning enhances the diversity of the population, which helps to improve its exploration ability. From the results of the 23 benchmark functions and three engineering design problems, the performance of the EOMSA was better than, or at least comparable with, other population-based algorithms mentioned in this paper. The EOMSA has a fast convergence speed and relatively high degree of stability. Additionally, it is much more accurate. To overcome the shortcomings of the algorithm, we can add other optimization strategies and improved methods to balance the exploration and exploitation ability. In future work, we will use the EOMSA to solve many

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

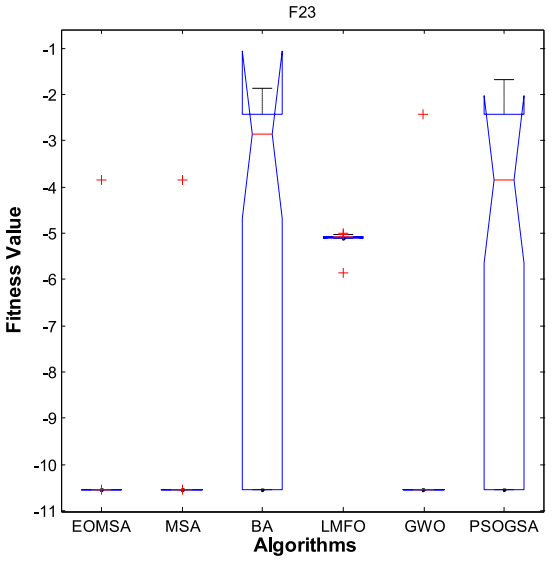

Fig. 46. $D = 30$: ANOVA test of the global minimum for f_{23}.

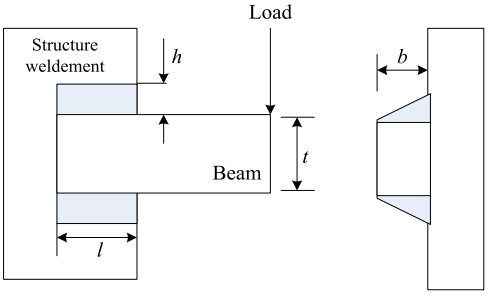

Fig. 47. Design parameters of the welded beam design problem.

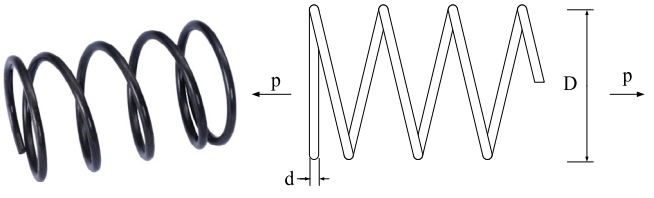

Fig. 48. Tension/compression spring design optimization.

classic combination optimization NP-hard problems, such as the traveling salesman problem, graph coloring problem and knapsack problem to improve the theoretical basis of the EOMSA and expand its application.

Acknowledgments

Funding: This work was supported by the National Science Foundation of China [grant numbers 61563008, 61463007]; and the Project of Guangxi University for Nationalities Science Foundation, China [grant numbers 2016GXNSFAA380264, 2018GXNSFAA138146]. We thank Maxine Garcia, Ph.D., from Liwen Bianji, Edanz Group China (www.liwenbianji.cn/ac) for editing the English text of a draft of this manuscript.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

Table 7
Comparison results for tension/compression spring design optimization.

Algorithm	Optimal values for variables			Optimal cost
	d	D	N	
EOMSA	0.22425	3.2486	0.22445	0.0099584
RO [28]	0.051370	0.349096	11.76279	0.0126788
SaC [45]	0.0521602	0.368158695	10.6484422	0.012669249
BGRA [24]	0.0516747	0.3563726	1.309229	0.012665237
IHS [35]	0.0511543	0.3498711	12.0764321	0.0126706
NM-PSO [56]	0.051620	0.355498	11.333272	0.0126706
CPSO [23]	0.051728	0.357644	11.24454	0.0126747
YYPO [43]	0.051705	0.35710	11.266	0.12665

Table 8
Comparison results for pressure vessel design optimization.

Algorithm	Optimal values for variables				Optimal cost
	T_s	T_h	R	L	
EOMSA	1.14602	0.566477	59.3791	37.8283	5879.7727
GSA [44]	1.125000	0.625000	55.9886598	84.4542025	8538.8359
GA [10]	0.937500	0.500000	48.329000	112.679000	6410.3811
WEO [27]	0.812500	0.437500	42.0984	176.6366	6059.7143
PSO-GA [19]	0.7781686	0.3846491	40.3196187	200.00	5885.3327736
Lagrangian Multiplier [26]	1.125000	0.625000	58.291000	43.6900000	7198.0428
Branch-bound [49]	1.125000	0.625000	47.700000	117.701000	8129.1036

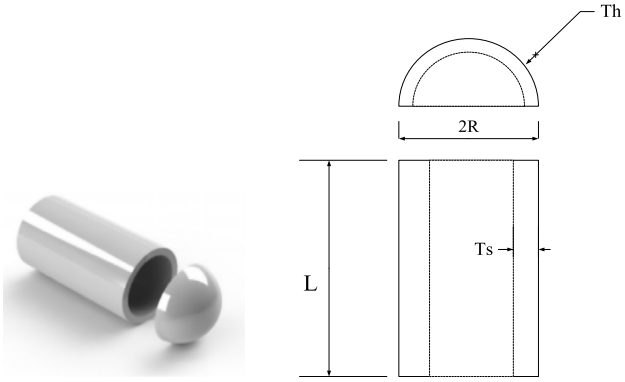

Fig. 49. Pressure vessel design optimization.

References

[1] E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm, Appl. Soft Comput. 41 (2016) 15–21.

[2] A. Baykasoğlu, Ş. Akpinar, Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems–Part 2: Constrained optimization, Appl. Soft Comput. 37 (2015) 396–415.

[3] Emer Bernal, Oscar Castillo, José Soria, Fevrier Valdez, Imperialist competitive algorithm with dynamic parameter adaptation using fuzzy logic applied to the optimization of mathematical functions, Algorithms 10 (1) (2017) 18.

[4] A.K. Bhandari, D. Kumar, A. Kumar, G.K. Singh, Optimal sub-band adaptive thresholding based edge preserved satellite image denoising using adaptive differential evolution algorithm, Neurocomputing 174 (2016) 698–721.

[5] H.R.E.H. Bouchekara, A.E. Chaib, M.A. Abido, R.A. El-Sehiemy, Optimal power flow using an improved colliding bodies optimization algorithm, Appl. Soft Comput. 42 (2016) 119–131.

[6] P.S. Callahan, 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) (1977) 3089–3097.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

[7] Camilo Caraveo, Fevrier Valdez, Oscar Castillo, A new optimization meta-heuristic algorithm based on self-defense mechanism of the plants with three reproduction operators, Soft Comput. 22 (15) (2018) 4907–4920.
[8] Oscar Castillo, Leticia Amador-Angulo, A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design, Inform. Sci. 460–461 (2018) 476–496.
[9] C.A. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Engrg. 191 (11) (2002) 1245–1287.
[10] C.A. Coello Coello, E. Mezura Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Adv. Eng. Inform. 16 (2002) 193–203.
[11] J.P. Cunningham, C.J. Moore, M.P. Zalucki, S.A. West, Learning, odour preference and flower foraging in moths, J. Exp. Biol. 207 (1) (2004) 87–94.
[12] K. Deb, Optimal design of a welded beam via genetic algorithms, AIAA J. 29 (11) (1991) 2013–2015.
[13] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm-anovel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166.
[14] R.J. Fan, P.E.T.E.R. Anderson, B. Hansson, Behavioural analysis of olfactory conditioning in the moth spodoptera littoralis, J. Exp. Biol. 200 (23) (1997) 2969–2976.
[15] Mantegna R.N. Fast, Accurate algorithm for numerical simulation of Levy stable stochastic processes, Phys. Rev. E 49 (5) (1994) 4677–4683.
[16] K.D. Frank, Impact of outdoor lighting on moths: an assessment, J. Lepidopterists' Soc. 42 (2) (1988) 63–93.
[17] K.D. Frank, C. Rich, T. Longcore, Effects of artificial night lighting on moths, in: Ecological Consequences of Artificial Night Lighting, vol. 13, Island Press, 2005, pp. 305–344.
[18] S. Garcia, D. Molina, et al., A study on the use of nonparametric tests for analyzing the evolutionary algorithms'behaviour: A case study on the CEC'2005 speciael session on real parameter optimization, J. Heuristics 15 (2009) 617–644.
[19] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016) 292–305.
[20] K.J. Gaston, J. Bennie, T.W. Davies, J. Hopkins, The ecological impacts of nighttime light pollution: a mechanistic appraisal, Biol. Rev. 88 (4) (2013) 912–927.
[21] B. González, F. Valdez, P. Melin, G. Prado-Arechiga, Fuzzy logic in the gravitational search algorithm for the optimization of modular neural networks in pattern recognition, Expert Syst. Appl. 42 (14) (2015) 5839–5847.
[22] N. Hansen, A. Auger, S. Finck, R. Ros, Real-Parameter Black-Box Optimization Benchmarking 2009 Experimental Setup, Institute National de Recherche en Informatique et en Automatique (INRIA), Rapports de Recherche RR-6828, Mar. 20, 2009.
[23] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (2007) 89–99.
[24] T.J. Hsieh, A bacterial gene recombination algorithm for solving constrained optimization problems, Appl. Math. Comput. 231 (2014) 187–204.
[25] F.-Z. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (2007) 340–356.
[26] B. Kannan, S.N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 116 (1994) 405.
[27] A. Kaveh, T. Bakhshpoori, Water Evaporation Optimization: A novel physically inspired optimization algorithm, Comput. Struct. 167 (2016) 69–85.
[28] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[29] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Eng. Comput. 27 (2010) 155–182.
[30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proc of the IEEE International Conference on Neural Networks. Perth, Austirlia, 1995, IV: 1942–1948.
[31] J.K. Kuang, S.S. Chen, Taguchi-aided search method for design optimization of engineering systems, Eng. Optim. 30 (1998) 1–23.
[32] Zhiming Li, Yongquan Zhou, Sen Zhang, Junmin Song, Lévy-Flight Moth-Flame Algorithm for Function Optimization and Engineering Design Problems, vol. 2016, 2016, p. 1423930, 22 pages.
[33] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Appl. Soft Comput. 10 (2010) 629–640.
[34] B. Mahdad, K. Srairi, Security constrained optimal power flow solution using new adaptive partitioning flower pollination algorithm, Appl. Soft Comput. 46 (2016) 501–522.
[35] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.
[36] R. Menzel, U. Greggers, M. Hammer, Functional organisation of appetitive learning and memory in a generalist pollinator, the honey bee, in: A.C. Lewis (Ed.), Insect Learning: Ecological and Evolutionary Perspectives, Chapman and Hall, London, 1993, pp. 79–125.
[37] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl. Based Syst. 89 (2015) 228–249.
[38] S. Mirjalili, S.Z.M. Hashim, A new hybrid pso-gsa algorithm for function optimization, in: Proceed of the Inter-National Conference on Computer and Information Application, ICCIA'10, IEEE, Tianjin, China, 2010, pp. 374–377.
[39] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[40] A.A.A. Mohamed, A.A. El-Gaafary, Y.S. Mohamed, A.M. Hemeida, Multi-objective states of matter search algorithm for TCSC-based smart controller design, Electr. Power Syst. Res. 140 (2016) 874–885.
[41] A.A.A. Mohamed, Y.S. Mohamed, A.A.M. El-Gaafary, et al., Optimal power flow using moth swarm algorithm, Electr. Power Syst. Res. 142 (2017) 190–206.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.

[42] Frumen Olivas, Fevrier Valdez, Oscar Castillo, Claudia I. González, Gabriela E. Martinez, Patricia Melin, Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems, Appl. Soft Comput. 53 (2017) 74–87.
[43] V. Punnathanam, P. Kotecha, Yin-Yang-pair optimization: A novel lightweight optimization algorithm, Eng. Appl. Artif. Intell. 54 (2016) 62–79.
[44] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inform. Sci. 179 (2009) 2232–2248.
[45] T. Ray, K.M. Liew, Society and civilization: An optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (4) (2003) 386–396.
[46] Luis Rodríguez, Oscar Castillo, José Soria, Patricia Melin, Fevrier Valdez, Claudia I. González, Gabriela E. Martinez, Jesus Soto, A fuzzy hierarchical operator in the grey wolf optimizer algorithm, Appl. Soft Comput. 57 (2017) 315–328.
[47] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (2013) 2592–2612.
[48] A. Sadollah, H. Eskandar, J.H. Kim, Water cycle algorithm for solving constrained multi-objective optimization problems, Appl. Soft Comput. 27 (2015) 279–298.
[49] E. Sandgren, Nonlinear integer and discrete programming in mechanical design, J. Mech. Des. (1988) 95–105.
[50] H.T. Skiri, M. Stranden, J.C. Sandoz, R. Menzel, H. Mustaparta, Associative learning of plant odorants activating the same or different receptor neurones in the moth heliothis virescens, J. Exp. Biol. 208 (4) (2005) 787–796.
[51] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European J. Oper. Res. 185 (3) (2008) 1155–1173.
[52] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[53] K. Tang, X. Yao, P.N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, Z. Yang, Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization, University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), Hefei, Anhui, China, Tech. Rep., 2007, [Online] Available.
[54] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. Biometr. 1 (6) (1944).
[55] X.-S. Yang, A new metaheuristic bat-inspired algorithm, Stud. Comput. Intell. 284 (2010) 65–74.
[56] E. Zahara, Y.T. Kao, Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Syst. Appl. 36 (2009) 3880–3886.

Please cite this article as: Q. Luo, X. Yang and Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Mathematics and Computers in Simulation (2018), https://doi.org/10.1016/j.matcom.2018.10.011.