Expert Systems With Applications 114 (2018) 563–577

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A modified whale optimization algorithm for large-scale global optimization problems

Yongjun Sun*, Xilu Wang, Yahuan Chen, Zujun Liu

State Key Laboratory of Integrated Service Networks, School of Telecommunication Engineering, Xidian University, No. 2 South Taibai Road, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 31 May 2018
Revised 28 July 2018
Accepted 14 August 2018
Available online 17 August 2018

Keywords:
Large-scale global optimization
Whale optimization algorithm
Lévy flight
Self-adaptive strategy
Quadratic interpolation

ABSTRACT

As a new and competitive population-based optimization algorithm, the Whale Optimization Algorithm (WOA) outperforms some other biological-inspired algorithms from the perspective of simplicity and efficiency. However, WOA will get stuck into local optima and degrade accuracy for large-scale global optimization (LSGO) problems. To address the issue, a modified Whale Optimization Algorithm (MWOA) is proposed for solving LSGO problems. In order to balance the exploration and exploitation abilities, a nonlinear dynamic strategy based on a cosine function for updating the control parameter is given. A Lévy-flight strategy is adopted to make the algorithm jump out of local optima. Moreover, a quadratic interpolation method is applied to the leader of the population, which enhances the local exploitation ability and improves the solution accuracy. MWOA is tested over 25 well-known benchmark functions with dimensions ranging from 100 to 1000. The experimental results demonstrate the superior performance of MWOA on LSGO, in terms of solution accuracy, convergence speed, and stability compared with other state-of-the-art optimization algorithms.

1. Introduction

Large-scale global optimization (LSGO) problems exist extensively in scientific research and engineering applications, such as aerospace design, machine learning, signal processing, and neural networks optimization (Li, Tang, Suganthan, & Yang, 2015; Lozano, Molina, & Herrera, 2011; Omidvar, Yang, Mei, Li, & Yao, 2017). These problems cannot be solved by traditional optimization algorithms because of high dimensions and complicated interaction among variables (Mohapatra, Das, & Roy, 2017). Against this background, meta-heuristic algorithms (MAs) have attracted great research interests and have been widely applied to practical problems.

The nature-inspired MAs make use of stochastic components without the gradient of the objective function, which is what we call the principle of trial-and-error (Boussaïd, Lepagnot, & Siarry, 2013). Specifically, MAs can be viewed as an iterative procedure, starting from an initial guess. New solutions are generated for each iteration according to the meta-heuristic operators. The subsequent step is to evaluate the fitness of each individual by the fitness function, and the fitness values are compared with those of the previous generations (Das & Suganthan, 2011). Also, an appropriate selection strategy is applied to determine which solutions will be preserved to the next generation (Yang, 2014, a). Generally, MAs with easy implementation are effective in providing optimal or near-optimal solutions for large scale optimization problems. Furthermore, MAs achieve superior performance in contrast with traditional optimization methods (Yang, 2014). This is why MAs are considered as a better option for LSGO. In the past decades, various MAs and their modifications have been proposed to tackle different optimization problems. MAs are mostly constituted by Evolutionary Algorithms and Swarm Intelligence (SI) algorithms (Boussaïd et al., 2013). Some of the competitive and popular Evolutionary Algorithms are Genetic Algorithm (GA) (Goldberg, 1989) and Differential Evolution (DE) (Das & Suganthan, 2011). SI algorithms mimic the swarm behavior of animal or insect societies, such as ant colony, bird schooling, fireflies flashing, wolve hunting, and whales prey. The most successful SI algorithms include Ant Colony Optimization (ACO) (Dorigo, Maniezzo, & Colorni, 1996), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Firefly Algorithm (FA) (Yang & Deb, 2010), Cuckoo Search (CS) (Yang, 2009), Grey Wolf Optimization (GWO) (Mirjalili, Mirjalili, & Lewis, 2014), Whale Optimization Algorithm (Mirjalili & Lewis, 2016, a) and their modifications.

When the dimension of optimization problems increases, especially for high dimensionality, the search space extends ex-

* Corresponding author.
E-mail addresses: yjsun@mail.xidian.edu.cn (Y. Sun), liuzujun@mail.xidian.edu.cn (Z. Liu).

https://doi.org/10.1016/j.eswa.2018.08.027
0957-4174/

ponentially, which results in “curse of dimensionality” (Howard, 1966). Based on the so-called “no free lunch” theorem (Wolpert & Macready, 1997), all optimization algorithms have shortcoming in some cases. Although many exsiting MAs can settle some practical issues, they are not capable of tackling LSGO problems. Hence, more efficient optimization algorithms with a better balance of exploitation and exploration are required (Ouyang, Gao, Li, & Kong, 2016). Such algorithms are mainly classified into two categories: Decomposition based algorithm and Non-Decomposition based algorithm, which are also called Cooperative Coevolution (CC) Algorithm and Hybrid Algorithm respectively (Mohapatra et al., 2017). The CC framework is developed by Potter and Jong to address high-dimensional problems (Potter, & Jong, 1994). Frans and Engelbrecht propose two PSO-based CC methods, i.e., CPSO-SK and CPSO–HK (Frans & Engelbrecht, 2004). Yang et al. introduce DE to CC framework and propose the DECC-G to sovle LSGO problems with dimensions of 500 and 1000 (Yang, Tang, & Yao, 2008a). Imanian et al. propose a hybrid algorithm named the velocity-based ABC for high dimensional continuous optimization problems (Imanian, Shiri, & Moradi, 2014). A hybrid simulated annealing and Nelder-Mead algorithm (SNMRUV) is applied to LSGO (Fouad, 2014), and its performance is good for some benchmark functions with dimensions of 500 and 1000 but not for all. The authors present a modified PSO with learning operation (IGPSO) in Ouyang, Gao, Li, and Kong (2017), which performs better for engineering design problems. An improved Opposition-Based Sine Cosine Algorithm is applied to global optimization (Elaziz, Oliva, & Xiong, 2017). Li et al. combine PSO with ABO (PS-ABC) for LSGO (Li, Wang, Yan, & Li, 2015) by utilizing the exploitation capability of PSO and the exploration ability of ABC, while PS-ABC still requires sufficiently large number of iterations. A Lévy flight trajectory-based WOA (LWOA) is introduced for global optimization problems with dimension up to 50 only (Ling, Zhou, & Luo, 2017). In these studies, MAs still suffer from several major drawbacks, such as low accuracy, premature convergence, and slow convergence speed when tackling LSCO problems.

This paper proposes a modified self-adaptive WOA (MWOA) based on a Lévy flight and a quadratic interpolation for LSGO problems. The Lévy flight helps WOA jump out of local optima owing to the frequent short-distance search step and occasionally the longer-distance search step, and the quadratic interpolation improves the solution accuracy by enhancing the exploitation ability. Furthermore, a self-adaptive control parameter achieves a better balance between the local exploitation and the global exploration. The rest of paper is organized as follows. In Section 2 a brief overview of WOA is provided. Section 3 introduces the strategies adopted by MWOA and describes the proposed MWOA in detail. Simulations and the discussion of results are shown in Section 4. Finally, the conclusions are given in Section 5.

2. Whale optimization algorithm

WOA is a population-based meta-heuristic algorithm and mimics the hunting mechanism of humpback whales. After humpback whales obtain the location of prey, they will dive into twelve meters down and trail spiral with bubbles upward to the surface. The location update is mainly divided into three phases: encircling prey, spiral bubble-net feeding maneuver, and search for prey (Mirjalili & Lewis, 2016).

2.1. Exploitation phase

During the exploitation process, humpback whales consider the current best location as the target prey (global optimum) and adjust their positions toward the global optimum according to two mechanisms: shrinking encircling (encircling prey) and spiral updating position (spiral bubble-net feeding maneuver). The mathematical model of the shrinking encircling mechanism is represented as follows:

$$\mathbf{D}=|C\,\boldsymbol{.}\,\mathbf{X}^*(t)-\mathbf{X}(t)| \tag{1}$$

$$\mathbf{X}(t+1)=\mathbf{X}^*(t)-A\,\boldsymbol{.}\,\mathbf{D} \tag{2}$$

where $\mathbf{X}$ is the position vector, $\mathbf{X}^*$ denotes the best solution obtained so far, which will be updated in each iteration if there is a better solution, t represents the current iteration, $|\bullet|$ denotes the absolute value operation, and $\bullet$ is an element-by-element multiplication. Here, A and C are two parameters, and they are calculated by

$$A=2a\,\boldsymbol{.}\,r-a \tag{3}$$

$$C=2\,\boldsymbol{.}\,r \tag{4}$$

where r is a random number in the interval [0,1], a decreases linearly from 2 to 0 throughout the iterations (in both exploration and exploitation phases) so that the shrinking encircling behavior can be achieved.

The spiral updating position is mathematically expressed by a spiral equation:

$$\mathbf{D}'=|\mathbf{X}^*(t)-\mathbf{X}(t)| \tag{5}$$

$$\mathbf{X}(t+1)=\mathbf{D}'\,\boldsymbol{.}\,e^{bl}\,\boldsymbol{.}\,\cos(2\pi l)+\mathbf{X}^*(t) \tag{6}$$

where $\mathbf{D}'$ represents the distance between the ith whale and the best solution obtained so far, l is a random number in $[-1,1]$ and b is a constant for defining the shape of a logarithmic spiral.

It is worth pointing out that the spiral-shaped path and the shrinking encircling are executed simultaneously when whales capture their prey. To imitate this behavior, each mechanism is implemented with 50% probability.

$$\mathbf{X}(t+1)=\begin{cases}\mathbf{X}^*(t)-A\,\boldsymbol{.}\,\mathbf{D} if p<0.5\\ \mathbf{D}'\,\boldsymbol{.}\,e^{bl}\,\boldsymbol{.}\,\cos(2\pi l)+\mathbf{X}^*(t) if p\geq 0.5\end{cases} \tag{7}$$

where p is a random number in [0,1].

2.2. Exploration phase

Here, a global search is developed to enhance the exploration ability. Its mathematical model is similar to Eq. (5) and Eq. (6), except for that a random search agent rather than the best agent is selected to guide the search. The random variable $|A|$ with the value larger than 1 or less than -1 is utilized to choose either exploration (search for prey) or exploitation (shrinking encircling mechanism) to update the position.

$$\mathbf{D}=|C\,\boldsymbol{.}\,\mathbf{X}_{rand}-\mathbf{X}| \tag{8}$$

$$\mathbf{X}(t+1)=\mathbf{X}_{rand}-A\,\boldsymbol{.}\,\mathbf{D} \tag{9}$$

where $\mathbf{X}_{rand}$ is a random position vector chosen from the current generation.

The pseudo code of WOA is presented in Algorithm 1. Note that WOA can smoothly convert between exploration and exploitation phases depending on only one parameter. Although WOA is competitive with some traditional MAs, it still suffer from some drawbacks for solving LSGO problems, such as low accuracy and premature convergence. Therefore, a modified WOA with a better trade-off between exploration and exploitation is proposed in the next section.

Algorithm 1 The whale optimization algorithm (WOA).

Initialization {
  initialize the whales population $\mathbf{X}_i$ ($i = 1, 2, 3..., n$)
  calculate the fitness of each search agent
  $\mathbf{X}*$ = the best search agent}
Main loop{
  while ($t <$ maximum number of iterations)
    for each search agent
      update *a, A, C, l*and*p*
      if1($p < 0.5$)
        if2($|A| < 1$)
        update the position of the current search agent by the Eq. (2)
        else if2($|A| \geq 1$)
        select a random search agent ($\mathbf{X}_{rand}$)
        update the position of the current search agent by the Eq. (9)
        end if2
      else if1 ($p \geq 0.5$)
        update the position of the current search agent by the Eq. (6)
      end if1
    end for
  check and repair duplicate genes to make sure each search agent is valid
  update $\mathbf{X}*$if there is a better solution
  $t = t + 1$
end while}
return $\mathbf{X}*$

Algorithm 2 The modified whale optimization algorithm (MWOA).

Initialization {
  initialize the whales population $\mathbf{X}_i$ ($i = 1, 2, 3..., n$)
  calculate the fitness of each search agent
  $\mathbf{X}*$ = the best search agent}
Main loop{
  while ($t <$ maximum number of iterations)
    for each search agent
      update *a* by the Eq. (19), update*A, C, l*, $p_1$and p_2
      if1($p_1 < 0.5$)
        if2($|A| < 1$)
        update the position of the current search agent by the Eq. (17)
        else if2($|A| \geq 1$)
        select a random search agent ($\mathbf{X}_{rand}$)
        update the position of the current search agent by the Eq. (9)
        end if2
      else if1 ($p_1 \geq 0.5$)
        if3($p_2 < 0.6$)
        update the position of the current search agent by the Eq. (6)
        else if3($p_2 \geq 0.6$)
        update the position of the current search agent by the Eq. (10)
      end if3
    end if1
  end for
  check and repair duplicate genes to make sure each search agent is valid
  update $\mathbf{X}^*$if there is a better solution
  $t = t + 1$
end while}
return $\mathbf{X}^*$

3. Modified whale optimization algorithm (MWOA)

In this section, WOA is modified by adjusting the control parameter and embedding other searching methods. In MWOA, three modifications are proposed and discussed in detail as follows.

3.1. Improvement based on quadratic interpolation (QI)

Although the basic WOA with the encircling mechanism and the spiral path is good at exploring the search space, it still requires improvement for tackling LSGO problems. Hence, in order to enhance the exploitation capability and improve the solution accuracy, a quadratic interpolation (QI) method is adopted in the MWOA. Mathematically, QI obtains the minimum point of the quadratic curve passing through three selected solutions in a n-dimensional space. Several successful attempts have been made over the years to integrate QI in MAs (Deep & Bansal, 2009; Deep & Das, 2008; Gupta, Deep, & Bansal, 2017; Singh & Agrawal 2016). Considered as a quadratic crossover operator, QI selects the best search agent $\mathbf{X}^* = (x_1^*, x_2^*, ..., x_n^*)$and the other two partners $\mathbf{Y} = (y_1, y_2, ..., y_n)$, $\mathbf{Z} = (z_1, z_2, ..., z_n)$ as three parents, and then generates an offspring (new solution) $\mathbf{X} = (x_1, x_2, ..., x_n)$ by the following equation.

$$x_i = 0.5 \cdot \frac{(y_i^2 - z_i^2) \cdot f(\mathbf{X}^*) + (z_i^2 - x_i^{*2}) \cdot f(\mathbf{Y}) + (x_i^{*2} - y_i^2) \cdot f(\mathbf{Z})}{(y_i - z_i) \cdot f(\mathbf{X}^*) + (z_i - x_i^*) \cdot f(\mathbf{Y}) + (x_i^* - y_i) \cdot f(\mathbf{Z})}, \quad \forall i = 1, 2, ..., n \tag{10}$$

where $f(\mathbf{X}*)$, $f(\mathbf{Y})$and $f(\mathbf{Z})$are the fitness values at $\mathbf{X}*$, $\mathbf{Y}$ and $\mathbf{Z}$ respectively, and i represents the ith dimension.

The current best search agent $\mathbf{X}*$ plays a guiding role in the quadratic crossover, which makes the search agents further explore the global optimal solution. QI is implemented in the exploitation phase for maintaining the diversity of population and enhancing the exploitation ability of MWOA. In this way, the exploitation phase of MWOA contains two components, the quadratic crossover and the spiral-shaped path. We take a uniformly distributed parameter to control the two components. When the probability is less than 0.6, the spiral-shaped-path method works as Eq. (6) in the original WOA. Otherwise, the quadratic crossover is impletmented to update the position of whales.

Table 1
Parameter settings of various algorithms.

Parameter settings	WOA	LWOA	EEGWO	BABC	MWOA
Population size	30	30	30	30	30
Max iterations	1000	1000	1000	100,000	1000
Control parameter	Linear	Linear	Nonlinear	Adaptive	Nonlinear

3.2. Improvement based on Lévy flight (LF)

The major issue of solving LSGO with MAs is that most of them converge prematurely toward local optima due to rapid reduction of diversity, and the original WOA is no exception. In the previous studies (Jensi & Jiji, 2016; Li, Zhou, Zhang, & Song, 2016; Salgotra, Singh, & Saha, 2017), the Lévy flight process is widely employed in MAs to prevent the solution from local optima and accelerate the convergence speed because of its efficient global search ability. Therefore, a Lévy flight is employed in MWOA to escape the local optima by promoting the population diversity.

The Lévy flight is a kind of non-Gaussian random process with step length following a Lévy distribution (Viswanathan et al., 1996). A simple power-law vision of the Lévy distribution is:

$$L(s) \sim |s|^{-1-\beta}, \; 0 < \beta \leq 2 \tag{11}$$

where β is an index, s is the step length of the Lévy flight. Mantegna's algorithm is applied to calculating s.

$$s = \mu / |\upsilon|^{1/\beta} \tag{12}$$

where μ and υ obey normal distribution, i.e.,

$$\mu \sim N(0, \sigma_\mu^2), \upsilon \sim N(0, \sigma_\upsilon^2) \tag{13}$$

$$\sigma_\mu = [\frac{\Gamma(1+\beta) \cdot \sin(\pi \cdot \beta/2)}{\Gamma((1+\beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}}]^{1/\beta} \tag{14}$$

$$\sigma_\upsilon = 1 \tag{15}$$

Table 2
Results and comparison of different algorithms on 25 benchmark functions with 100*D*.

Function		WOA	LWOA	EEGWO	BABC	MWOA
f_1	Mean	2.76 E−161	**0.00 E+00**	5.11 E−114	1.80 E−16	**0.00 E+00**
	Std	1.23 E−160	**0.00 E+00**	2.08 E−113	8.06 E−16	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_2	Mean	3.09 E−105	1.36 E−204	6.26 E−72	2.18 E−15	**0.00 E+00**
	Std	1.38 E−104	0.00 E+00	2.00 E−71	9.75 E−15	**0.00 E+00**
	Rank	3	2	4	5	1
f_3	Mean	5.08 E+03	1.23 E−265	5.16 E+00	2.72 E−34	**0.00 E+00**
	Std	1.26 E+04	0.00 E+00	1.50 E+01	6.81 E−34	**0.00 E+00**
	Rank	5	2	4	3	1
f_4	Mean	2.84 E−24	1.74 E−168	3.79 E−35	2.18 E−03	**1.38** E−**19**
	Std	1.04 E−23	0.00 E+00	1.32 E−34	6.71 E−03	**4.20** E−**19**
	Rank	4	2	3	5	1
f_5	Mean	2.19 E+01	3.46 E+01	9.82 E+01	1.60 E−01	**0.00 E+00**
	Std	3.93 E+01	4.59 E+01	1.46 E−01	5.63 E−01	**0.00 E+00**
	Rank	3	4	5	2	1
f_6	Mean	6.26 E−02	5.71 E−02	4.95 E+00	2.99 E−07	**0.00 E+00**
	Std	1.43 E−01	4.01 E−02	2.26 E+00	1.34 E−06	**0.00 E+00**
	Rank	4	3	5	2	1
f_7	Mean	2.97 E−04	2.52 E−04	4.67 E−04	1.49 E−04	**0.00 E+00**
	Std	5.62 E−04	3.01 E−04	4.96 E−04	9.13 E−05	**0.00 E+00**
	Rank	4	3	5	2	1
f_8	Mean	7.13 E−168	**0.00 E+00**	4.59 E−118	3.67 E−38	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	1.53 E−117	1.27 E−37	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_9	Mean	4.62 E−160	**0.00 E+00**	4.78 E−115	8.67 E−36	**0.00 E+00**
	Std	2.07 E−159	**0.00 E+00**	2.12 E−114	3.85 E−35	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{10}	Mean	4.63 E−01	5.11 E−01	6.69 E−01	5.47 E−01	**2.50** E−**01**
	Std	2.39 E−01	2.03 E−01	1.72 E−03	2.50 E−01	**1.33** E−**04**
	Rank	2	3	5	4	1
f_{11}	Mean	9.52 E−165	**0.00 E+00**	2.10 E−113	5.27 E−11	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	6.41 E−113	1.82 E−10	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{12}	Mean	8.49 E−165	**0.00 E+00**	8.62 E−118	5.14 E−15	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	2.16 E−117	2.30 E−14	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{13}	Mean	1.68 E+03	1.41 E−52	4.01 E+03	9.35 E−15	**0.00 E+00**
	Std	3.02 E+02	6.32 E−52	1.06 E+03	3.92 E−14	**0.00 E+00**
	Rank	4	2	5	3	1
f_{14}	Mean	−4.15 E+04	−4.07 E+04	−3.92 E+04	−4.19 E+04	**−4.01 E+04**
	Std	7.82 E+02	2.36 E+03	4.29 E+03	1.13 E+02	**8.17** E−**01**
	Rank	3	4	5	2	1
f_{15}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{16}	Mean	2.84 E−15	**8.88** E−**16**	2.31 E−15	1.60 E−15	**8.88** E−**16**
	Std	1.81 E−15	**0.00 E+00**	1.79 E−15	1.46 E−15	**0.00 E+00**
	Rank	5	1.5	4	3	1.5
f_{17}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{18}	MEan	6.10 E−04	6.38 E−04	3.02 E−02	5.49 E−08	**0.00 E+00**
	Std	6.80 E−04	6.64 E−04	1.84 E−02	2.46 E−07	**0.00 E+00**
	Rank	3	4	5	2	1
f_{19}	Mean	1.97 E−02	2.85 E−02	1.12 E+00	1.35 E−32	**0.00 E+00**
	Std	2.81 E−02	5.92 E−02	8.31 E−01	2.81 E−48	**0.00 E+00**
	Rank	3	4	5	2	1
f_{20}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{21}	Mean	5.58 E−112	1.29 E−201	6.53 E−71	1.39 E−14	**0.00 E+00**
	Std	1.60 E−111	0.00 E+00	1.99 E−70	6.21 E−14	**0.00 E+00**
	Rank	3	2	4	5	1
f_{22}	Mean	6.25 E−04	**0.00 E+00**	3.13 E−04	3.13 E−04	**0.00 E+00**
	Std	1.28 E−03	**0.00 E+00**	9.62 E−04	9.62 E−04	**0.00 E+00**
	Rank	5	1.5	3.5	3.5	1.5
f_{23}	Mean	−7.83 E+01	−7.83 E+01	−7.64 E+01	−7.83 E+01	**−7.83 E+01**
	Std	4.14 E−03	1.69 E−03	3.74 E+00	4.17 E−07	**2.19** E−**07**
	Rank	4	3	5	2	1
f_{24}	Mean	6.37 E−161	**0.00 E+00**	8.48 E−119	1.87 E−11	**0.00 E+00**
	Std	2.85 E−160	**0.00 E+00**	2.96 E−118	8.36 E−11	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{25}	Mean	−6.21 E+00	−6.30 E+00	−6.26 E+00	**−6.30 E+00**	−6.30 E+00
	Std	1.85 E−01	3.26 E−03	6.77 E−02	**5.72** E−**15**	3.32 E−06
	Rank	5	3	4	1	2
Ave rank		3.48	2.48	4.18	3.42	1.44
Overall rank		3	2	5	4	1

Table 3
R Esults and comparison of diff Er Ent algorithms on 25 b Enchmark functions with 300*D*.

Function		WOA	LWOA	EEGWO	BABC	MWOA
f_1	Mean	1.12 E−162	**0.00 E+00**	1.94 E−114	2.63 E−25	**0.00 E+00**
	Std	4.97 E−162	**0.00 E+00**	8.42 E−114	6.42 E−25	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_2	Mean	5.73 E−107	6.19 E−211	1.98 E−69	1.84 E−09	**0.00 E+00**
	td	2.56 E−106	0.00 E+00	8.72 E−69	8.18 E−09	**0.00 E+00**
	Rank	3	2	4	5	1
f_3	Mean	2.06 E+06	1.88 E−209	2.86 E+04	2.06 E−22	**0.00 E+00**
	Std	2.59 E+06	0.00 E+00	8.75 E+04	9.19 E−22	**0.00 E+00**
	Rank	5	2	4	3	1
f_4	Mean	4.71 E−18	5.64 E−171	5.07 E−34	1.99 E−02	**0.00 E+00**
	Std	2.11 E−17	0.00 E+00	2.19 E−33	5.42 E−02	**0.00 E+00**
	Rank	4	2	3	5	1
f_5	Mean	1.77 E+01	1.09 E+01	2.97 E+02	1.37 E+01	**1.64** E−**19**
	Std	6.56 E+01	1.24 E+01	2.91 E−01	3.34 E+01	**7.35** E−**19**
	Rank	4	2	5	3	1
f_6	Mean	1.26 E−01	3.69 E−01	2.18 E+01	1.30 E−23	**0.00 E+00**
	Std	1.39 E−01	5.28 E−01	8.90 E+00	5.78 E−23	**0.00 E+00**
	Rank	3	4	5	2	1
f_7	Mean	4.23 E−04	1.75 E−04	2.39 E−04	1.61 E−04	**0.00 E+00**
	Std	3.69 E−04	2.56 E−04	2.09 E−04	1.37 E−04	**0.00 E+00**
	Rank	5	3	4	2	1
f_8	Mean	4.93 E−164	**0.00 E+00**	3.61 E−115	1.29 E−25	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	1.61 E−114	4.84 E−25	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_9	Mean	6.35E−162	**0.00 E+00**	1.31 E−112	1.49 E−24	**0.00 E+00**
	Std	2.66 E−161	**0.00 E+00**	4.52 E−112	6.44 E−24	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{10}	Mean	7.17 E−01	6.10 E−01	6.70 E−01	6.25 E−01	**2.50** E−**01**
	Std	3.20 E−01	1.52 E−01	4.92 E−03	2.33 E−01	**1.07** E−**04**
	Rank	5	2	4	3	1
f_{11}	Mean	3.91 E−157	**0.00 E+00**	1.25 E−110	1.18 E−06	**0.00 E+00**
	Std	1.75 E−156	**0.00 E+00**	4.14 E−110	4.51 E−06	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{12}	Mean	2.44 E−167	**0.00 E+00**	4.03 E−114	1.15 E−26	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	1.74 E−113	3.76 E−26	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{13}	Mean	4.61 E+03	1.23 E+03	1.46 E+04	1.75 E−07	**0.00 E+00**
	Std	7.53 E+02	2.59 E+03	1.10 E+03	6.00 E−07	**0.00 E+00**
	Rank	4	3	5	2	1
f_{14}	Mean	−1.25 E+05	−1.25 E+05	−1.22 E+05	−1.17 E+05	−1.26 E+05
	Std	2.79 E+03	1.13 E+03	9.13 E+03	1.56 E+04	4.62 E+00
	Rank	3	2	4	5	1
f_{15}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{16}	Mean	2.13 E−15	**8.88** E−**16**	1.73 E−15	1.14 E−14	**8.88** E−**16**
	Std	1.74 E−15	**0.00 E+00**	1.46 E−15	3.97 E−14	**0.00 E+00**
	Rank	4	1.5	3	5	1.5
f_{17}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{18}	Mean	2.46 E−04	4.47 E−04	3.02 E−02	2.87 E−24	**2.04** E−**32**
	Std	2.75 E−04	5.25 E−04	2.53 E−02	9.94 E−24	**6.72** E−**32**
	Rank	3	4	5	2	1
f_{19}	Mean	7.61 E−02	9.67 E−02	5.34 E+00	2.47 E−08	**0.00 E+00**
	Std	1.05 E−01	1.14 E−01	3.21 E+00	8.72 E−08	**0.00 E+00**
	Rank	3	4	5	2	1
f_{20}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{21}	Mean	8.79 E−113	3.95 E−204	2.34 E−53	1.20 E−05	**0.00 E+00**
	Std	2.51 E−112	0.00 E+00	1.05 E−52	4.40 E−05	**0.00 E+00**
	Rank	3	2	4	5	1
f_{22}	Mean	9.38 E−04	**0.00 E+00**	9.38 E−04	1.56 E−04	**0.00 E+00**
	Std	1.47 E−03	**0.00 E+00**	1.47 E−03	6.99 E−04	**0.00 E+00**
	Rank	4.5	1.5	4.5	3	1.5
f_{23}	Mean	−7.83 E+01	−7.83 E+01	−7.56 E+01	−7.24 E+01	**−7.83 E+01**
	Std	2.18 E−03	2.11 E−03	4.78 E+00	1.15 E+01	**4.92** E−**06**
	Rank	3	2	4	5	1
f_{24}	Mean	1.42 E−167	**0.00 E+00**	1.18 E−116	3.46 E−16	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	5.26 E−116	1.55 E−15	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{25}	Mean	−1.87 E+01	−1.89 E+01	−1.86 E+01	**−1.89 E+01**	−1.89 E+01
	Std	6.98 E−01	2.17 E−02	5.43 E−01	**2.88** E−**13**	1.17 E−06
	Rank	4	3	5	1	2
Ave rank		3.50	2.32	3.98	3.76	1.44
Overall rank		3	2	5	4	1

Table 4
Results and comparison of different algorithms on 25 benchmark functions with 500*D*.

Function		WOA	LWOA	EEGWO	BABC	MWOA
f_1	Mean	1.72 E−162	**0.00 E+00**	2.30 E−112	5.46 E−21	**0.00 E+00**
	Std	7.70 E−162	**0.00 E+00**	1.02 E−111	2.41 E−20	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_2	Mean	1.21 E−109	2.63 E−199	3.62 E−70	6.49 E−10	**0.00 E+00**
	Std	3.87 E−109	0.00 E+00	1.24 E−69	2.90 E−09	**0.00 E+00**
	Rank	3	2	4	5	1
f_3	Mean	9.09 E+06	1.09 E−197	8.24 E+05	9.03 E−19	**0.00 E+00**
	Std	7.81 E+06	0.00 E+00	1.98 E+06	3.24 E−18	**0.00 E+00**
	Rank	5	2	4	3	1
f_4	Mean	5.15 E−19	1.71 E−161	2.73 E−37	5.70 E−01	**0.00 E+00**
	Std	2.30 E−18	7.66 E−161	3.73 E−37	2.52 E+00	**0.00 E+00**
	Rank	4	2	3	5	1
f_5	Mean	3.62 E+01	1.36 E+02	4.96 E+02	4.99 E+01	**4.06** E−**19**
	Std	1.10 E+02	2.13 E+02	5.59 E−01	1.13 E+02	**1.00** E−**18**
	Rank	2	4	5	3	1
f_6	Mean	2.64 E−01	5.10 E−01	4.23 E+01	4.49 E−19	**0.00 E+00**
	Std	2.75 E−01	5.39 E−01	1.75 E+01	2.01 E−18	**0.00 E+00**
	Rank	3	4	5	2	1
f_7	Mean	2.84 E−04	1.03 E−04	3.83 E−04	1.28 E−04	**0.00 E+00**
	Std	4.40 E−04	1.16 E−04	4.44 E−04	1.12 E−04	**0.00 E+00**
	Rank	4	2	5	3	1
f_8	Mean	3.85 E−162	**0.00 E+00**	8.81 E−115	1.15 E−22	**0.00 E+00**
	Std	1.72 E−161	**0.00 E+00**	3.10 E−114	4.17 E−22	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_9	Mean	1.45 E−158	**0.00 E+00**	1.39 E−111	5.52 E−19	**0.00 E+00**
	Std	6.49 E−158	**0.00 E+00**	6.08 E−111	2.41 E−18	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{10}	Mean	8.93 E−01	6.32 E−01	6.73 E−01	5.96 E−01	**3.25** E−**01**
	Std	2.23 E−01	1.31 E−01	7.75 E−03	2.35 E−01	**2.31** E−**01**
	Rank	5	3	4	2	1
f_{11}	Mean	3.52 E−158	**0.00 E+00**	3.69 E−112	2.50 E−08	**0.00 E+00**
	Std	1.58 E−157	**0.00 E+00**	1.08 E−111	1.12 E−07	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{12}	Mean	3.84 E−164	**0.00 E+00**	1.09 E−112	2.23 E−11	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	4.44 E−112	9.97 E−11	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{13}	Mean	8.32 E+03	5.00 E+03	2.41 E+04	2.20 E−07	**0.00 E+00**
	Std	9.92 E+02	4.22 E+03	2.31 E+03	8.62 E−07	**0.00 E+00**
	Rank	4	3	5	2	1
f_{14}	Mean	−2.09 E+05	−2.07 E+05	−1.99 E+05	−1.84 E+05	**−2.09 E+05**
	Std	1.95 E+03	4.39 E+03	2.17 E+04	2.88 E+04	**2.40 E+00**
	Rank	2	3	4	5	1
f_{15}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{16}	Mean	1.42 E−15	**8.88** E−**16**	2.66 E−15	2.40 E−14	**8.88** E−**16**
	Std	1.30 E−15	**0.00 E+00**	1.82 E−15	4.57 E−14	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{17}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{18}	Mean	3.46 E−04	5.23 E−04	5.47 E−02	4.72 E−07	**3.46** E−**32**
	Std	4.97 E−04	6.06 E−04	5.01 E−02	2.11 E−06	**6.13** E−**32**
	Rank	3	4	5	2	1
f_{19}	Mean	9.16 E−02	3.12 E−01	1.06 E+01	1.54 E−05	**1.03** E−**28**
	Std	8.89 E−02	4.80 E−01	6.92 E+00	6.88 E−05	**1.79** E−**28**
	Rank	3	4	5	2	1
f_{20}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{21}	Mean	1.69 E−107	1.95 E−210	5.76 E−03	1.39 E−04	**0.00 E+00**
	Std	5.17 E−107	0.00 E+00	2.57 E−02	6.20 E−04	**0.00 E+00**
	Rank	3	2	5	4	1
f_{22}	Mean	7.82 E−04	**0.00 E+00**	6.25 E−04	1.56 E−04	**0.00 E+00**
	Std	1.39 E−03	**0.00 E+00**	1.28 E−03	6.99 E−04	**0.00 E+00**
	Rank	5	1.5	4	3	1.5
f_{23}	Mean	−7.83 E+01	−7.83 E+01	−7.53 E+01	−7.83 E+01	**−7.83 E+01**
	Std	9.53 E−03	3.35 E−02	4.18 E+00	4.45 E−03	**5.11** E−**07**
	Rank	3	4	5	2	1
f_{24}	Mean	5.32 E−170	**0.00 E+00**	8.40 E−117	5.22 E−26	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	3.44 E−116	1.71 E−25	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{25}	Mean	−3.13 E+01	−3.14 E+01	−3.11 E+01	**−3.15 E+01**	−3.15 E+01
	Std	4.82 E−01	1.83 E−01	8.60 E−01	**8.31** E−**08**	1.57 E−05
	Rank	4	3	5	1	2
Ave rank		3.32	2.52	4.20	3.52	1.44
Overall rank		3	2	5	4	1

Table 5
Results and comparison of different algorithms on 25 benchmark functions with 1000*D*.

Function		WOA	LWOA	EEGWO	BABC	MWOA
f_1	Mean	3.66 E−159	**0.00 E+00**	9.22 E−110	5.19 E−16	**0.00 E+00**
	Std	1.63 E−158	**0.00 E+00**	4.12 E−109	2.30 E−15	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_2	Mean	3.02 E−107	1.81 E−193	2.87 E−70	9.12 E−11	**0.00 E+00**
	Std	1.34 E−106	0.00 E+00	9.11 E−70	4.07 E−10	**0.00 E+00**
	Rank	3	2	4	5	1
f_3	Mean	5.75 E+07	6.87 E−167	5.23 E+06	1.77 E−18	**0.00 E+00**
	Std	4.64 E+07	0.00 E+00	8.26 E+06	6.33 E−18	**0.00 E+00**
	Rank	5	2	4	3	1
f_4	Mean	1.76 E−22	7.56 E−165	1.25 E−32	4.31 E−02	**0.00 E+00**
	Std	4.75 E−22	0.00 E+00	5.48 E−32	1.32 E−01	**0.00 E+00**
	Rank	4	2	3	5	1
f_5	Mean	2.72 E+01	5.18 E+02	9.94 E+02	3.18 E+02	**1.38** E−**17**
	Std	6.39 E+01	4.86 E+02	1.06 E+00	3.16 E+02	**4.20** E−**17**
	Rank	2	4	5	3	1
f_6	Mean	7.73 E−01	7.53 E−01	8.33 E+01	5.44 E−14	**0.00 E+00**
	Std	1.51 E+00	7.71 E−01	3.11 E+01	2.43 E−13	**0.00 E+00**
	Rank	4	3	5	2	1
f_7	Mean	4.41 E−04	1.42 E−04	3.65 E−04	1.54 E−04	**0.00 E+00**
	Std	6.97 E−04	1.91 E−04	4.32 E−04	8.52 E−05	**0.00 E+00**
	Rank	5	2	4	3	1
f_8	Mean	8.29 E−171	**0.00 E+00**	4.58 E−114	3.13 E−21	**0.00 E+00**
	Std	0.00 E+00	**0.00 E+00**	1.93 E−113	1.23 E−20	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_9	Mean	1.14 E−150	**0.00 E+00**	1.50 E−111	4.66 E−12	**0.00 E+00**
	Std	5.11 E−150	**0.00 E+00**	6.57 E−111	1.96 E−11	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{10}	Mean	8.82 E−01	6.63 E−01	6.77 E−01	5.95 E−01	**3.35** E−**01**
	Std	2.01 E−01	1.49 E−01	1.77 E−02	1.08 E+00	**3.64** E−**01**
	Rank	5	3	4	2	1
f_{11}	Mean	3.08 E−157	**0.00 E+00**	8.44 E−108	7.45 E−06	**0.00 E+00**
	Std	1.36 E−156	**0.00 E+00**	2.89 E−107	2.99 E−05	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{12}	Mean	2.62 E−156	**0.00 E+00**	3.03 E−113	1.11 E−13	**0.00 E+00**
	Std	1.17 E−155	**0.00 E+00**	9.04 E−113	4.49 E−13	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{13}	Mean	1.51 E+04	7.61 E+03	5.10 E+04	3.34 E−07	**0.00 E+00**
	Std	1.31 E+03	7.79 E+03	1.12 E+04	8.46 E−07	**0.00 E+00**
	Rank	4	3	5	2	1
f_{14}	Mean	−4.18 E+05	−4.14 E+05	−3.99 E+05	−3.74 E+05	**−4.13 E+05**
	Std	1.94 E+03	6.37 E+03	3.69 E+04	6.07 E+04	**1.29 E+01**
	Rank	2	3	4	5	1
f_{15}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{16}	Mean	1.24 E−15	**8.88** E−**16**	3.02 E−15	5.65 E−14	**8.88** E−**16**
	Std	1.09 E−15	**0.00 E+00**	1.79 E−15	1.56 E−13	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{17}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{18}	Mean	2.51 E−04	4.49 E−04	3.01 E−02	2.45 E−11	**4.13** E−**28**
	Std	3.17 E−04	5.35 E−04	2.35 E−02	1.09 E−10	**0.00 E+00**
	Rank	3	4	5	2	1
f_{19}	Mean	2.12 E−01	4.64 E−01	1.79 E+01	2.26 E−10	**6.65** E−**28**
	Std	2.73 E−01	4.85 E−01	1.15 E+01	1.01 E−09	**2.10** E−**27**
	Rank	3	4	5	2	1
f_{20}	Mean	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Std	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**	**0.00 E+00**
	Rank	3	3	3	3	3
f_{21}	Mean	1.35 E−110	1.91 E−206	1.06 E−69	3.08 E−07	**0.00 E+00**
	Std	3.68 E−110	0.00 E+00	4.26 E−69	1.14 E−06	**0.00 E+00**
	Rank	3	2	4	5	1
f_{22}	Mean	3.13 E−04	**0.00 E+00**	7.82 E−04	1.56 E−04	**0.00 E+00**
	Std	9.62 E−04	**0.00 E+00**	1.39 E−03	6.99 E−04	**0.00 E+00**
	Rank	4	1.5	5	3	1.5
f_{23}	Mean	−7.83 E+01	−7.83 E+01	−7.69 E+01	−7.27 E+01	**−7.83 E+01**
	Std	2.94 E−02	6.77 E−03	3.00 E+00	1.16 E+01	**1.60** E−**06**
	Rank	3	2	4	5	1
f_{24}	Mean	7.00 E−159	**0.00 E+00**	8.06 E−115	3.87 E−14	**0.00 E+00**
	Std	3.14 E−158	**0.00 E+00**	3.47 E−114	1.73 E−13	**0.00 E+00**
	Rank	3	1.5	4	5	1.5
f_{25}	Mean	−6.30 E+01	−6.30 E+01	−6.20 E+01	−6.30 E+01	**−6.30 E+01**
	Std	8.08 E−02	4.93 E−02	1.15 E+00	2.95 E−03	**7.68** E−**06**
	Rank	4	3	5	2	1
Ave rank		3.36	2.40	4.12	3.72	1.40
Overall rank		3	2	5	4	1

Table 6
The results of wilcoxon's rank sum test for all functions with 100*D*.

Function		MWOA Vs.			
		WOA	LWOA	EEGWO	BABC
f_1	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_2	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_3	*p*-value	8.01 E−09	3.51 E−07	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_4	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_5	*p*-value	6.80 E−08	6.80 E−08	6.80 E−08	6.80 E−08
	h-value	1	1	1	1
f_6	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	0.34
	h-value	1	1	1	0
f_7	*p*-value	0.23	0.23	0.00	0.54
	h-value	0	0	1	0
f_8	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_9	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{10}	*p*-value	7.90 E−08	3.94 E−07	6.80 E−08	0.03
	h-value	1	1	1	1
f_{11}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{12}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{13}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_{14}	*p*-value	0.01	0.00	0.00	0.04
	h-value	1	1	1	1
f_{15}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{16}	*p*-value	1.95 E−02	NaN	4.68 E−05	0.01
	h-value	1	0	1	1
f_{17}	*p*-value	NaN	NaN	NaN	0.34
	h-value	0	0	0	0
f_{18}	*p*-value	5.28 E−08	5.28 E−08	5.28 E−08	0.00
	h-value	1	1	1	1
f_{19}	*p*-value	3.98 E−08	3.98 E−08	3.98 E−08	0.12
	h-value	1	1	1	0
f_{20}	*p*-value	NaN	NaN	NaN	0.34
	h-value	0	0	0	0
f_{21}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	0.00
	h-value	1	1	1	1
f_{22}	*p*-value	0.16	NaN	0.08	0.34
	h-value	0	0	0	0
f_{23}	*p*-value	1.06 E−07	1.06 E−07	2.22 E−07	1.54 E−06
	h-value	1	1	1	1
f_{24}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{25}	*p*-value	7.90 E−08	6.80 E−08	6.80 E−08	6.66 E−08
	h-value	1	1	1	1

Table 7
The results of wilcoxon's rank sum test for all functions with 300*D*.

Function		MWOA Vs.			
		WOA	LWOA	EEGWO	BABC
f_1	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_2	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_3	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_4	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_5	*p*-value	6.79 E−08	6.79 E−08	6.79 E−08	6.79 E−08
	h-value	1	1	1	1
f_6	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	2.57 E−05
	h-value	1	1	1	1
f_7	*p*-value	0.14	0.27	1.44 E−04	0.01
	h-value	0	0	1	1
f_8	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_9	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{10}	*p*-value	6.53 E−04	0.09	0.11	0.67
	h-valu E	1	0	0	0
f_{11}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{12}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{13}	*p*-value	8.01 E09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_{14}	*p*-value	2.99 E−08	2.99 E−08	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_{15}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{16}	*p*-value	0.0398	NaN	0.0043	3.6587 E−04
	h-value	1	0	1	1
f_{17}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{18}	*p*-value	4.44 E−08	4.44 E−08	4.44 E−08	0.02
	h-value	1	1	1	1
f_{19}	*p*-value	4.46 E−08	4.46 E−08	4.46 E−08	0.69
	h-value	1	1	1	1
f_{20}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{21}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	9.43 E−06
	h-value	1	1	1	1
f_{22}	*p*-value	0.02	NaN	0.02	0.34
	h-value	1	0	1	0
f_{23}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_{24}	*p*-value	3.84 E−10	NaN	3.84 E−10	3.84 E−10
	h-value	1	0	1	1
f_{25}	*p*-value	4.21 E−08	4.21 E−08	4.21 E−08	7.72 E−08
	h-value	1	1	1	1

A step size avoiding the Lévy flight jumping out of the design domain is adopted (Yang & Deb, 2013). It is defined by:

$$\textbf{Levy}{=}random(size(\mathbf{D})) \oplus \mathbf{L}(\beta) \sim 0.01\mu/|\upsilon|^{1/\beta}(\mathbf{X}_i - \mathbf{X}^*) \tag{16}$$

where *size(**D**)* is the scale of the problem, $\oplus$ denotes entry-wise multiplications, $\mathbf{X}_i$ is the *i*th solution vector.

Due to the infinite variance of Lévy distribution, the Lévy flight executes the long-distance movement occasionally for promoting the exploration ability, while the short-distance movement is performed for enhancing the exploitation ability. Obviously, this merit can ensure that MAs jump out of local optima. In MWOA, the shrinking encircling mechanism is replaced by a Lévy flight in order to explore the search space more efficiently. The new position is updated according to the following rule.

$$\mathbf{X}(t+1) = \mathbf{X}(t) + 1/sqrt(t)\,\boldsymbol{.}\,sign(rand - 0.5) \oplus \textbf{Levy} \tag{17}$$

where *1/sqrt(t)* is a parameter related to the current iteration number *t* and *sqrt*() denotes the square root operation. In this regard, a larger range of search movement can be executed during the early stage while a smaller one is employed in the later period. *Sign(rand-0.5)* represents a sign function with only three values −1, 0, 1, which makes the search more random. The exploration phase of MWOA is summarized as follows:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}(t) + 1/sqrt(t)\,\boldsymbol{.}\,sign(rand - 0.5) \oplus \textbf{Levy}\, if\, p < 0.5 \\ \mathbf{D}'\,\boldsymbol{.}\,e^{bl}\,\boldsymbol{.}\cos(2\pi l) + \mathbf{X}^*(t)\, if\, p \geq 0.5 \end{cases} \tag{18}$$

3.3. Nonlinear control parameter strategy

All MAs have two essential components: exploitation and exploration. The key to achieve good performance relies on whether the algorithm can make a perfect harmony between exploitation and exploration. In the standard WOA, the coefficient *A* has significant influence on balancing exploitation and exploration. As stated above, the value $|A| > 1$ forces whales to explore the search

Table 8
The results of wilcoxon's rank sum test for all functions with 500*D*.

Function		MWOA Vs.			
		WOA	LWOA	EEGWO	BABC
f_1	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_2	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_3	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_4	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_5	*p*-value	6.79 E−08	6.79 E−08	6.79 E−08	6.79 E−08
	h-value	1	1	1	1
f_6	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_7	*p*-value	0.02	0.32	2.47 E−04	0.00
	h-value	1	0	1	1
f_8	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_9	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{10}	*p*-value	0.00	0.27	0.27	0.07
	h-value	1	0	0	0
f_{11}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{12}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{13}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_{14}	*p*-value	2.99 E−08	1.10 E−06	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_{15}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{16}	*p*-value	0.0195	NaN	4.97 E−06	9.21 E−04
	h-value	1	0	1	1
f_{17}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{18}	*p*-value	5.64 E−08	5.64 E−08	5.64 E−08	7.72 E−07
	h-value	1	1	1	1
f_{19}	*p*-value	5.28 E−08	5.28 E−08	5.28 E−08	3.79 E−04
	h-value	1	1	1	1
f_{20}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{21}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	3.31 E−06
	h-value	1	1	1	1
f_{22}	*p*-value	0.16	NaN	0.16	0.16
	h-value	0	0	0	0
f_{23}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_{24}	*p*-value	3.84 E−10	NaN	3.84 E−10	3.84 E−10
	h-value	1	0	1	1
f_{25}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	4.01 E−04
	h-value	1	1	1	1

Table 9
The results of wilcoxon's rank sum test for all functions with 1000*D*.

Function		MWOA Vs.			
		WOA	LWOA	EEGWO	BABC
f_1	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_2	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_3	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_4	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_5	*p*-value	6.77 E−08	6.77 E−08	6.77 E−08	6.77 E−08
	h-value	1	1	1	1
f_6	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	9.30 E−04
	h-value	1	1	1	1
f_7	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	8.01 E−09
	h-value	1	1	1	1
f_8	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_9	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{10}	*p*-value	0.00	0.00	0.00	0.29
	h-value	1	1	1	0
f_{11}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{12}	*p*-value	8.01 E−09	NaN	8.01 E−09	8.01 E−09
	h-value	1	0	1	1
f_{13}	*p*-value	8.01 E−09	8.01 E−09	7.98 E−09	8.01 E−09
	h-value	1	1	1	1
f_{14}	*p*-value	4.38 E−05	3.45 E−05	3.80 E−05	6.75 E−08
	h-value	1	1	1	1
f_{15}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{16}	*p*-value	0.28	NaN	0.34	0.25
	h-value	0	0	0	0
f_{17}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{18}	*p*-value	6.80 E−08	6.80 E−08	6.80 E−08	0.01
	h-value	1	1	1	1
f_{19}	*p*-value	6.73 E−08	6.73 E−08	6.73 E−08	2.93 E−07
	h-value	1	1	1	1
f_{20}	*p*-value	NaN	NaN	NaN	NaN
	h-value	0	0	0	0
f_{21}	*p*-value	8.01 E−09	8.01 E−09	8.01 E−09	3.31 E−06
	h-value	1	1	1	1
f_{22}	*p*-value	0.18	NaN	0.12	0.28
	h-value	0	0	0	0
f_{23}	*p*-value	0.12	0.12	0.12	0.31
	h-value	0	0	0	0
f_{24}	*p*-value	3.84 E−10	NaN	3.84 E−10	3.84 E−10
	h-value	1	0	1	1
f_{25}	*p*-value	6.74 E−08	6.74 E−08	6.74 E−08	0.03
	h-value	1	1	1	0

space and the value $|A| < 1$ forces whales to attack towards the prey (exploitation). The parameter a decreases linearly from 2 to 0 over the course of iterations, which directly affects the coefficient A. Unfortunately, the linearly decreasing parameter a can neither accurately reflect nor well adapt to the complicated and nonlinear search process. Taking this into consideration, a nonlinear control parameter is used in MWOA to make an obvious difference to the solution quality. MWOA employs a cosine function for updating a during iterations:

$$a = 2 \cdot cos(t/Max\_iter) \qquad (19)$$

where $Max_iter$is the maximum number of iterations.

The pseudo code of MWOA is shown in Algorithm 2.

4. Experimental results and discussion

In this section, MWOA is evaluated by scalable benchmark functions. The other four state-of-the-art algorithms are selected for comparison.

4.1. Experimental setting

The performance of MWOA is compared with the WOA (Mirjalili & Lewis, 2016), the Lévy flight trajectory-based WOA (LWOA) (Ling et al., 2017), the exploration-enhanced GWO (EEGWO) (Heidari & Pahlavani, 2017) and the adaptive BABC (BABC) (Banharnsakun, Achalakul, & Sirinaovakul, 2011). For the sake of fairness, the parameter settings for different algorithms are obtained from the original papers respectively, as shown in Table 1.

4.2. Benchmark functions

A series of high-dimensional benchmark functions listed in the appendix are utilized to test the performance of the algorithms. These functions can be divided into unimodal and multimodal functions. The unimodal functions $f_1 \sim f_{13}$ have only one global optimum so they are used to investigate the exploitation capability, and the multimodal functions $f_{14} \sim f_{25}$ with more than two

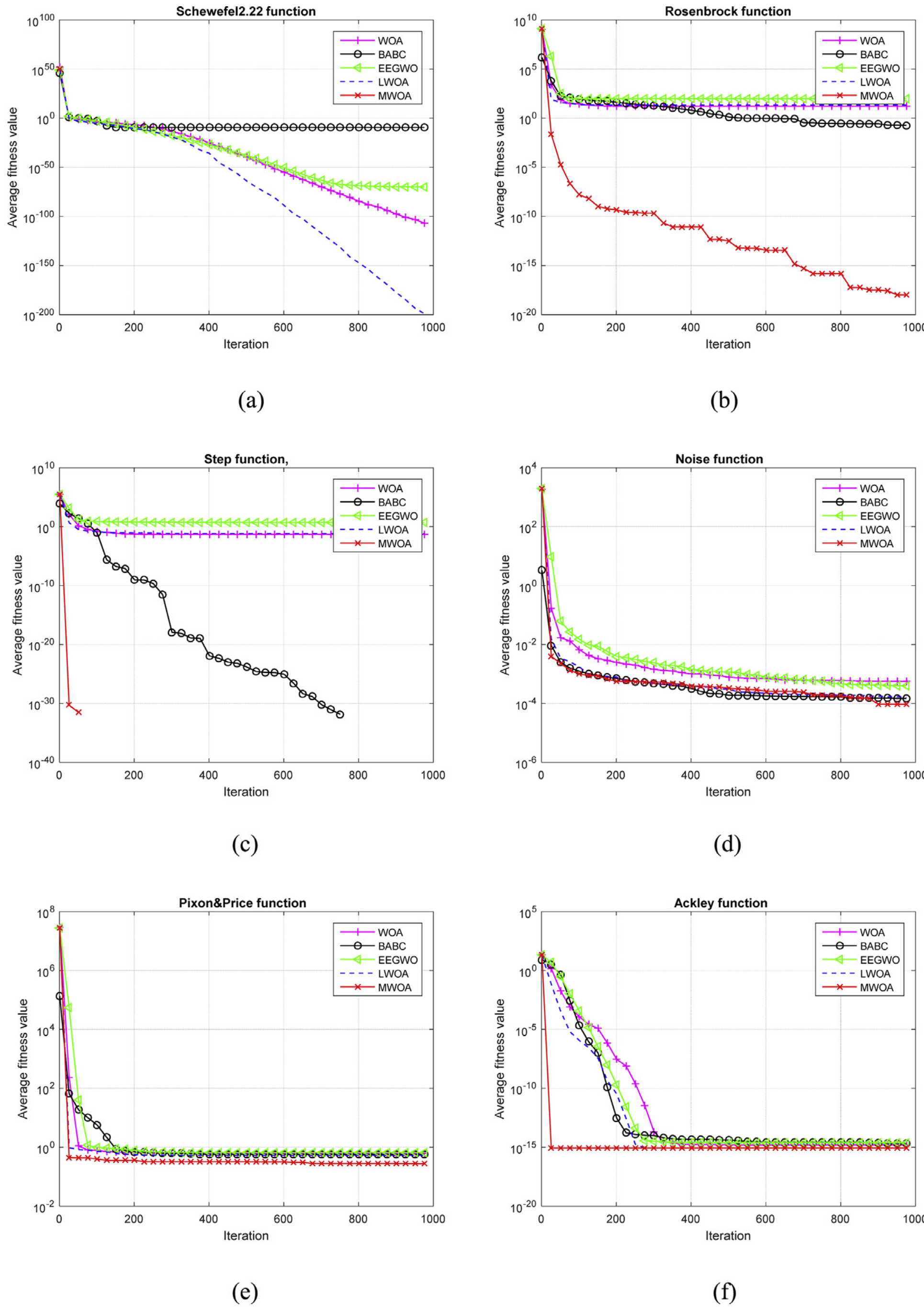

Fig. 1. The convergence curve of average fitness value on selected functions with 100*D*.

local optima are utilized to evaluate the exploration ability. In the appendix, *D* denotes the number of function dimensions.

4.3. Comparison and analysis of simulation results

To evaluate the performance of the proposed algorithm on LSGO problems, the dimension of the functions is set to 100, 300, 500, and 1000 respectively. All algorithms run 20 times independently for each benchmark function. Based on the statistical analysis, the mean and standard deviation (Std) of 20 running results for the benchmark functions are recorded in Tables 2–5. Wilcoxon's test, shown in Tables 6–9, is employed to estimate whether there is a statistical significant difference between MWOA and the compared algorithms. Moreover, a comparison of the convergence rate is illustrated in Figs. 1–2.

4.3.1. Comparison of simulation results

The simulation results of benchmark functions with dimensions $D = 100$, 300, 500, and 1000 are shown in Tables 2–5 respectively, where the best mean and the best Std of the benchmark functions are highlighted in bold. Tables 2–5 also rank the performance of algorithms by"tied rank". Specifically, the algorithms are ranked from the best to the worst according to the mean values. Algorithms with the same performance are assigned to the average

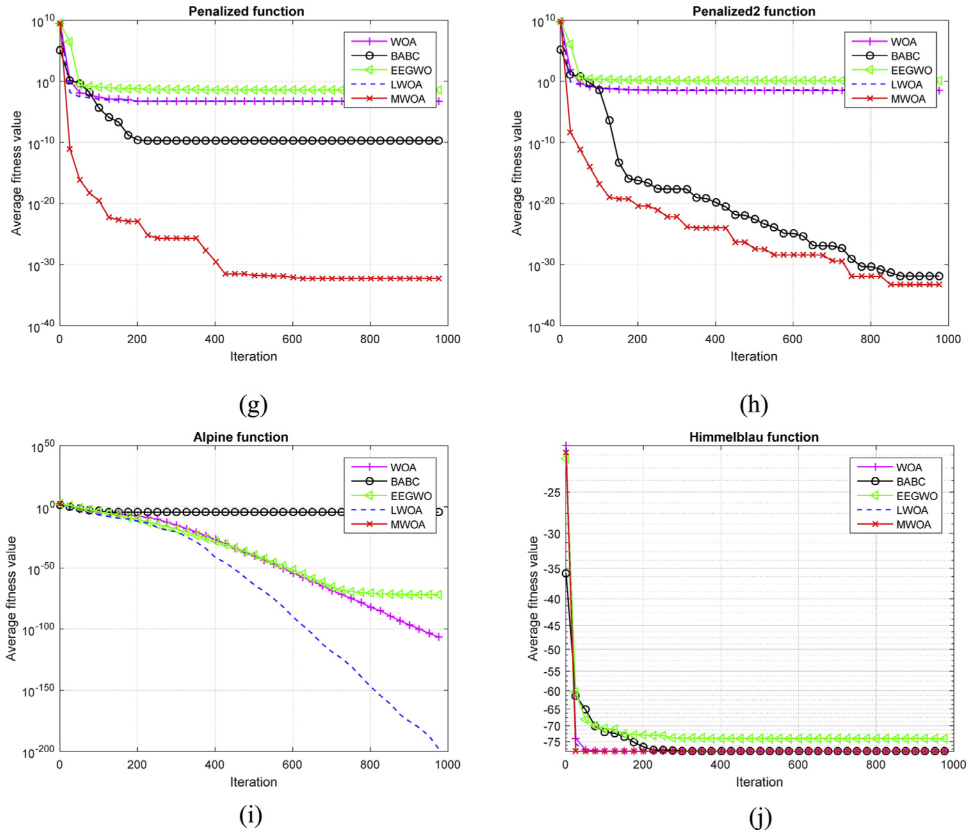

Fig. 1. Continued

rank. Moreover, Tables 2–5 offer the average and overall rank of each algorithm for 25 benchmark functions. Through the results of different dimensions, it is obvious that MWOA invariably obtains the highest rank, followed by LWOA, WOA, BABC, and finally EEGWO. Moreover, from Tables 2–5 it is clear that MWOA finds the global optimal value for most of the benchmark functions, irrespective of their dimensionality.

From the results for the unimodal functions f_1~f_{13}, MWOA performs best and is proven to have the good exploitation ability. MWOA converges to the optimal solution for the unimodal functions except f_4 and f_{10} as the quadratic interpolation enhances the exploitation capability. For the multimodal functions f_{14}~f_{25}, MWOA outperforms the other compared algorithms for all functions except f_{25}. For f_{25} MWOA ranks second and BABC gets the best performance with $D = 100$, 300 and 500. The results also verify that the Lévy flight can help MWOA effectively avoid getting into local optima. The results for the multimodal functions reveal that MWOA provides sufficient global search to explore the large search space.

In conclusion, the results show that MWOA has a powerful exploitation ability to dig in promising areas and efficient exploration capability to search the whole space. Therefore, MWOA can effectively solve LSGO problems.

4.3.2. Wilcoxon's rank sum test results

To estimate the statistical significance difference between the algorithms, Wilcoxon's non-parametric statistical test (Derrac, García, Molina, & Herrera, 2011) is employed at a 5% significance level. The test uses two sets of samples and then obtains the *p*-values and *h*-values, which can serve as an indicator of the significance level. Specifically, if the *p*-values are less than 5% or the *h*-values equal to 1, the corresponding algorithms are considered to have statistically significant superiority in solving LSGO. The results of comparing MWOA with the other four algorithms on the benchmark functions with $D = 100$, 300, 500, 1000 are presented in Tables 6–9 respectively.

It is noted that all algorithms converge to the global optimal value for f_{15}, f_{17}, and f_{20}, which leads to the *p*-values to be NaN. In the following discussion we do not consider the three functions with the same performance. For the 100*D* functions, MWOA has a significant difference in comparison with WOA and EEGWO, except for f_7(not suitable for EEGWO) and f_{22}. Thers is also a significant difference between MWOA and LWOA except f_7. Compared with BABC the proposed algorithm has a significant difference for 18 functions. Similar conclusions can be drawn from Tables 7–9 when the dimensions are 300, 500 and 1000. Generally, MWOA has a significant difference compared with WOA, EEGWO, and BABC for most functions with different dimensions. Apart from the functions with same performance, Tables 7–9 show that there is also a significant difference between MWOA and LWOA for higher dimension functions.

4.3.3. Convergence rate comparison

The averaged convergence curves over 20 independent runs are drawn in Figs. 1–2. The function dimensions are set to 100 and 1000 respectively. The selected functions f_2, f_5, f_6, f_7, f_{10}, f_{16}, f_{18}, f_{19}, $f_{21}$and f_{23} are Schewefel 2.22 function, Rosenbrock function, Step function, Noise function, Pixon&Price function, Ackley function, Penalized function, Penalized 2 function, Alpine function, and Himmelblau function respectively. The first five functions are unimodal functions and the last five are multimodal functions. As depicted in Figs. 1–2, MWOA not only converges quickly towards the global optimal solution but also achieves higher accuracy. On the contrary, the compared algorithms converge easily to the local optima for most functions, and the convergence speed will be very slow when they converge toward the global optimal solution.

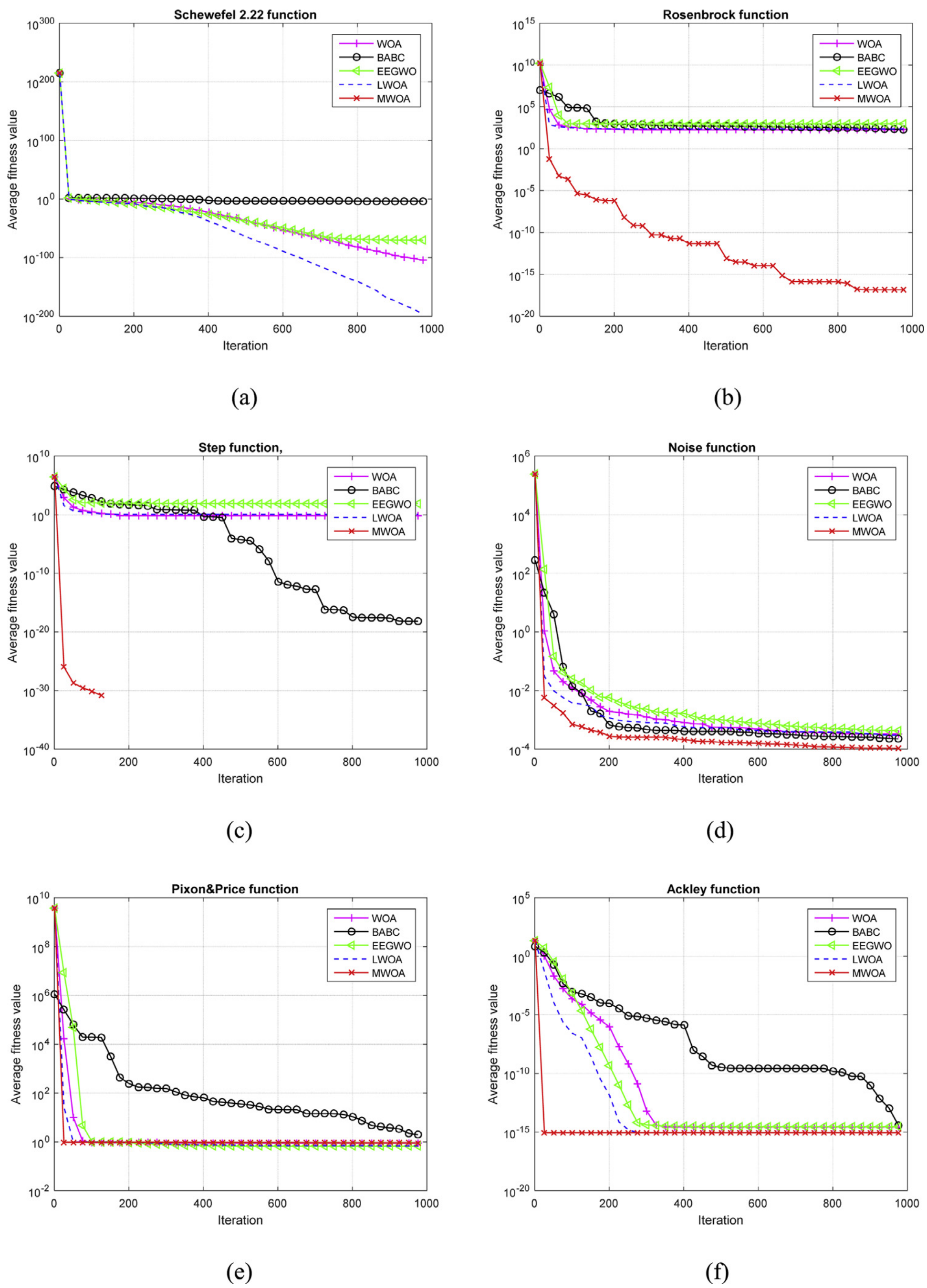

Fig. 2. The convergence curve of mean fitness value on some selected functions with 1000*D*.

From Figs. 1–2(a) and (c), it is obvious that MWOA converges very fast and reaches optimal results with less than 10 iterations for Schewefel2.22 and Step functions. The global optimal value of Rosenbrock function lies in such a narrow valley that it is difficult for optimization algorithms to converge to an ideal solution. According to the red curves in Figs. 1–2(b), MWOA obtains a near-global optimal solution, whereas the other four competitors plunge into local optima at the beginning. As shown in Figs. 1–2(d) and (e), the performances of MWOA for Noise function and Pixon&Price function are not remarkable among the algorithms, but MWOA still gets the best performance.

For Ackley, Alpine, and Himmelblau functions, MWOA acheves the fastest convergence speed and the best solution amog all algorithms. According to Figs. 1–2(g) and (h), MWOA searches much deeper into the promising space on Penalized and Penalized2 functions, and other algoriths quickly fall into the local optima except BABC for Penalized2 function.

5. Conclusion

The original WOA suffers from the premature convergence and low accuracy for tackling LSGO problems. In this research, a modified WOA is proposed to overcome the shortcomings. A Lévy flight with an adaptive step size is introduced to help the algorithm jump out of local optima, and a quadratic interpolation is adopted in the exploitation phase for digging deeper into the search space.

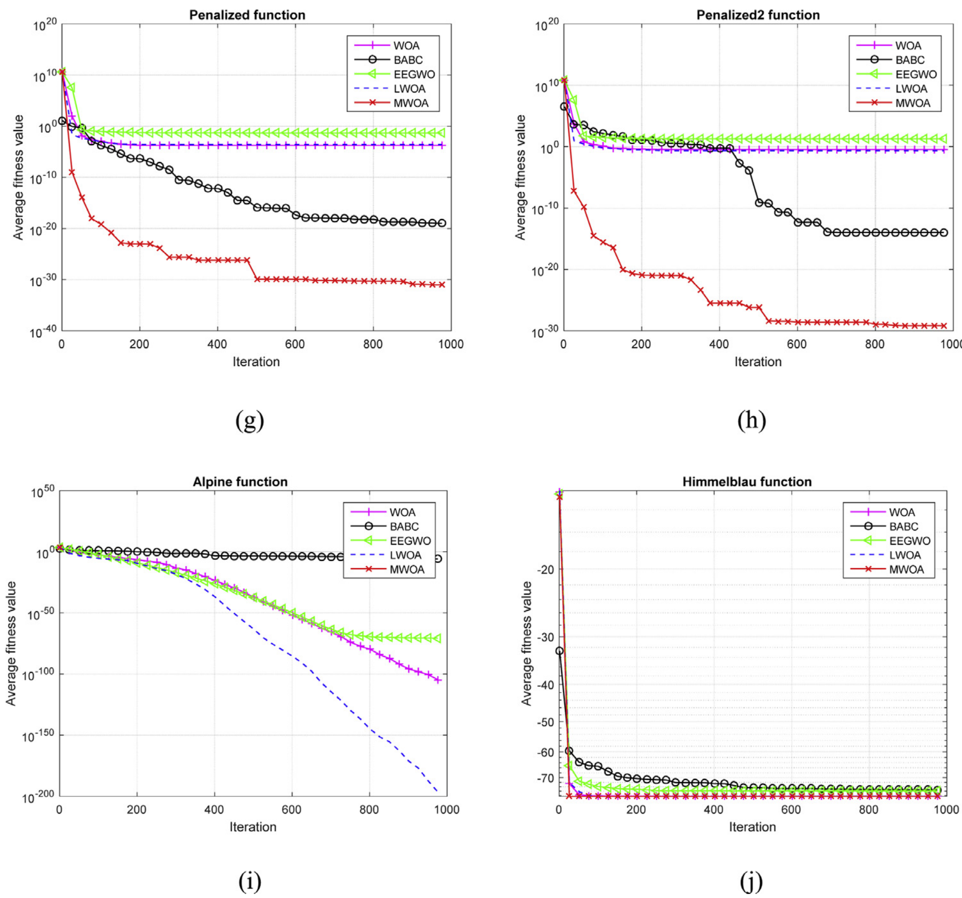

Fig. 2. Continued

The whole search process is controlled by a nonlinear control strategy, which helps the exploration and exploitation run more smoothly and improves the performance of MWOA.

Multiple well-known benchmark functions are used to evaluate the performance of the algorithms. The experimental results validate that MWOA has a superior performance on LSGO problems in terms of convergence rate and accuracy. MWOA is a promising optimization algorithm and may be applied to the practical and engineering problems for further research.

Author Contribution

Yongjun Sun: Conceptualization; Data curation; Formal analysis; Roles/Writing - original draft; Supervision; Methodology.

Xilu Wang: Data curation; Software simulation; Visualization; Validation.

Yahuan Chen: Data curation; Software simulation.

Zujun Liu: Funding acquisition; Writing - review & editing.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China [grant number 61571340], and the program of Introducing Talents of Discipline to Universities [grant number B0803].

Appendix

Function name	Equation	Search range	fmin
Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$	[−100,100]	0
Schewefel 2.22	$f_2 = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	[−10,10]	0
Schewefel 1.2	$f_3 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	[−100,100]	0
Schewefel 2.21	$f_4 = \max\{\lvert x_i\rvert, 1 \le i \le D\}$	[−100,100]	0
Rosenbrock	$f_5 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[−30,30]	0
Step	$f_6 = \sum_{i=1}^{D} \lfloor x_i + 0.5 \rfloor^2$	[−100,100]	0
Noise	$f_7 = \sum_{i=1}^{D} ix_i^4 + random(0, 1)$	[−1.28,1.28]	0
Cigar	$f_8 = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^6$	[−100,100]	0
Tablet	$f_9 = 10^6 \cdot x_1^2 + \sum_{i=2}^{D} x_i^6$	[−1,1]	0
Pixon &Price	$f_{10} = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_{i-1})^2$	[−10,10]	0
Elliptic	$f_{11} = \sum_{i=2}^{D} (10^6)^{(i-1)/(n-1)} \cdot x_i^2$	[−100,100]	0
Sum squares	$f_{12} = \sum_{i=2}^{D} ix_i^2$	[−10,10]	0
Zakharov	$f_{13} = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5ix_i)^2 + (\sum_{i=1}^{D} 0.5ix_i)^4$	[−5,10]	0
Schwefel 2.26	$f_{14} = \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i\rvert})$	[−500,500]	$-418.9826D$
Rastrigin	$f_{15} = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[−5.12,5.12]	0
Ackley	$f_{16} = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2})$ $-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	[−32,32]	0
Griewank	$f_{17} = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[−60,60]	0
Penalized	$f_{18} = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$	[−50,50]	0
Penalized 2	$f_{19} = 0.1\{\sin(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + 10\sin^2(3\pi x_{i+1})]$ $+(x_n - 1)^2\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	[−50,50]	0
Weierstrass's	$f_{20} = \sum_{i=1}^{D}[\sum_{k=0}^{k_{\max}} a^k\cos(2\pi b^k(x_i + 0.5))$ $-n\sum_{k=0}^{k_{\max}} a^k\cos(\pi b^k)], k_{\max} = 20, a = 0.5, b = 3$	[−50,50]	0
Alpine	$f_{21} = \sum_{i=1}^{D} \lvert x_i\sin(x_i) + 0.1x_i\rvert$	[−10,10]	0
Generalized schaffer	$f_{22} = 0.5 + ((\sin(\sum_{i=1}^{D} x_i^2))^2 - 0.5)$ $\cdot(1 + 0.001(\sum_{i=1}^{D} x_i^2))^{-2}$	[−100,100]	0
Himmelblau	$f_{23} = \frac{1}{n}\sum_{i=1}^{D}(x_i^4 - 16x_i^2 + 5x_i)$	[−5,5]	−78.33236
Bohachevsky	$f_{24} = \sum_{i=1}^{D-1}[x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i)$ $-0.4\cos(4\pi x_{i+1}) + 0.7]$	[−15,15]	0
Cosine mixture	$f_{25} = -[-0.1\sum_{i=1}^{D}\cos(5\pi x_i) - \sum_{i=1}^{D} x_i^2]$	[−1,1]	$-0.1D$

References

Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2011). The best-so-far selection in artificial bee colony algorithm. *Applied Soft Computing Journal, 11*(2), 2888–2901.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237*(237), 82–117.

D Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation, 15*(1), 4–31.

Deep, K., & Bansal, J. C. (2009). Hybridization of particle swarm optimization with quadratic approximation. *Opsearch, 46*(1), 3–24.

Deep, K., & Das, K. N. (2008). Quadratic approximation based hybrid genetic algorithm for function optimization. *Applied Mathematics & Computation, 203*(1), 86–98.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm & Evolutionary Computation, 1*(1), 3–18.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 26*(1), 29–41.

Elaziz, M. A., Oliva, D., & Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications, 90*, 484–500.

Fouad, A. (2014). Hybrid simulated annealing and nelder-mead algorithm for solving large-scale global optimization problems. *International Journal of Research in Computer Science, 4*(3), 1–11.

Frans, V. D. B., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 225–239.

Goldberg, D. E. (1989). Genetic algorithm in search optimization and machine learning. *Addison Wesley Professional*.

Gupta, K., Deep, K., & Bansal, J. C. (2017). Improving the local search ability of spider monkey optimization algorithm using quadratic approximation for unconstrained optimization. *Computational Intelligence, 33*(2), 210–240.

Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with lévy flight for optimization tasks. *Applied Soft Computing, 60*, 115–134.

Howard, R. A. (1966). Dynamic programming. *Manage Sci, 12*(5), 317–348.

Imanian, N., Shiri, M. E., & Moradi, P. (2014). Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems. *Engineering Applications of Artificial Intelligence, 36*, 148–163.

J Jensi, R., & Jiji, G. W. (2016). An enhanced particle swarm optimization with levy flight for global optimization. *Applied Soft Computing, 43*(C), 248–261.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In IEEE International Conference on Neural Networks*, 1995.* Proceedings*: Vol.4* (pp. 1942–1948). IEEE. vol. 4.

Li, X., Tang, K., Suganthan, P. N., & Yang, Z. (2015). Editorial for the special issue of information sciences journal, (isj) on "nature-inspired algorithms for large scale global optimization. *Information Sciences An International Journal, 316*(C), 437–439.

Li, Z., Wang, W., Yan, Y., & Li, Z. (2015). PS–ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems. *Expert Systems with Applications, 42*(22), 8881–8895.

Li, Z., Zhou, Y., Zhang, S., & Song, J. (2016). Lévy-flight moth-flame algorithm for function optimization and engineering design problems. *Mathematical Problems in Engineering, 2016*, 1–22.

Ling, Y., Zhou, Y., & Luo, Q. (2017). Lévy flight trajectory-based whale optimization algorithm for global optimization. *IEEE Access, 5*(99), 6168–6186.

Lozano, M., Molina, D., & Herrera, F. (2011). Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. *Soft Computing, 15*(11), 2085–2087.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*(3), 46–61.

Mohapatra, P., Das, K. N., & Roy, S. (2017). A modified competitive swarm optimizer for large scale optimization problems. *Applied Soft Computing, 59*, 340–362.

Omidvar, M. N., Yang, M., Mei, Y., Li, X., & Yao, X. (2017). Dg2: A faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Transactions on Evolutionary Computation, 21*(6), 929–942.

Ouyang, H. B., Gao, L. Q., Li, S., & Kong, X. Y. (2016). Improved global-best-guided particle swarm optimization with learning operation for global optimization problems. *Appl Soft Comput, 52*(C), 987–1008.

Ouyang, H. B., Gao, L. Q., Li, S., & Kong, X. Y. (2017). Improved global-best-guided particle swarm optimization with learning operation for global optimization problems. *Applied Soft Computing, 52*(C), 987–1008.

Potter, M. A., & Jong, K. A. D. (1994). A cooperative coevolutionary approach to function optimization. *Third Parallel Problem Solving Form Nature, 866*, 249–257.

Salgotra, R., Singh, U., & Saha, S. (2017). New cuckoo search algorithms with enhanced exploration and exploitation properties. *Expert Systems with Applications, 95*, 384–420.

Singh, D., & Agrawal, S. (2016). Self organizing migrating algorithm with quadratic interpolation for solving large scale global optimization problems. *Applied Soft Computing, 38*, 1040–1048.

Viswanathan, G. M., Afanasyev, V., Buldyrev, S. V., Murphy, E. J., Prince, P. A., & Stanley, H. E. (1996). Lévy flight search patterns of wandering albatrosses. *Nature, 381*(6581), 413–415.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. *Mathematics, 5792*, 169–178.

Yang, X. S. (2014). Swarm intelligence based algorithms: A critical analysis. *Evolutionary Intelligence, 7*(1), 17–28.

Yang, X. S., & Deb, S. (2010). Cuckoo Search via Levey Flights. Nature & Biologically Inspired Computing, 2009. In *NaBIC 2009.* World Congress on (pp. 210–214).

Yang, X. S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research, 40*(6), 1616–1624.

Yang, Z., Tang, K., & Yao, X. (2008a). Large scale evolutionary optimization using cooperative coevolution. *Information Sciences, 178*(15), 2985–2999.