Applied Soft Computing Journal 76 (2019) 16–30

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Multi-strategy ensemble grey wolf optimizer and its application to feature selection

Qiang Tu [a], Xuechen Chen [a,*], Xingcheng Liu [a,b,*]

[a] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, Guangdong, China
[b] School of Information Science, Xinhua College of Sun Yat-sen University, Guangzhou, Guangdong, China

HIGHLIGHTS

- A multi-strategy ensemble GWO is proposed to boost the precision and efficiency of the original GWO.
- A parameter self-adjusting strategy is utilized to balance the exploitation and exploration of the proposed MEGWO.
- Wilcoxons signed-rank test and performance profile are used to investigate the significance of the MEGWO.
- Feature selection is employed to evaluate the effectiveness of MEGWO on real-world applications.

ARTICLE INFO

Article history:
Received 26 July 2018
Received in revised form 9 November 2018
Accepted 29 November 2018
Available online 10 December 2018

Keywords:
Intelligent simulation
Function optimization
Grey wolf optimizer
Multi-strategy ensemble
Feature selection

ABSTRACT

To overcome the limitation of single search strategy of grey wolf optimizer (GWO) in solving various function optimization problems, we propose a multi-strategy ensemble GWO (MEGWO) in this paper. The proposed MEGWO incorporates three different search strategies to update the solutions. Firstly, the enhanced global-best lead strategy can improve the local search ability of GWO by fully exploiting the search space around the current best solution. Secondly, the adaptable cooperative strategy embeds one-dimensional update operation into the framework of GWO to provide a higher population diversity and promote the global search ability. Thirdly, the disperse foraging strategy forces a part of search agents to explore a promising area based on a self-adjusting parameter, which contributes to the balance between the exploitation and exploration. We conducted numerical experiments based on various functions form CEC2014. The obtained results are compared with other three modified GWO and seven state-of-the-art algorithms. Furthermore, feature selection is employed to investigate the effectiveness of MEGWO on real-world applications. The experimental results show that the proposed algorithm which integrate multiple improved search strategies, outperforms other variants of GWO and other algorithms in terms of accuracy and convergence speed. It is validated that MEGWO is an efficient and reliable algorithm not only for optimization of functions with different characteristics but also for real-world optimization problems.

1. Introduction

In recent decades, nature-inspired intelligent simulation optimization algorithms (ISOAs) by mimicking the laws of natural evolution, physical rules or biological behaviours have thrived. Optimization problems brought up in complex systems are usually characterized by non-linearity, non-differentiability, multimodality, discontinuity, inseparability and inequality constraints. Traditional mathematical programming methods are no longer efficient in resolving such problems [1]. In contrast, ISOAs with good self-learning ability and efficient search mechanism become competitive alternatives.

In 1980s and 1990s, with the rapid development of modern computing and data storage technology, some classical and powerful ISOAs have been proposed and extensively investigated, including Simulated Annealing algorithm (SA) [2], Genetic Algorithm (GA) [3], Back Propagation Neural Network (BPNN) [4], Differential Evolution (DE) [5], Particle Swarm Optimization (PSO) [6], etc. Since the 21st century, the inspirations from nature have continued to spawn the rapid development of ISOAs. During recent years, not only traditional ISOAs have been further studied, but also a variety of novel heuristic algorithms appear successively, such as: Firefly Algorithm (FA) [7], Harmony Search (HS) [8], Bat Algorithm (BA) [9], Cuckoo Search (CS) [10], Bacteria Foraging Algorithm (BFA) [11], Shuffled Frog Leaping Algorithm (SFLA) [12], Artificial Bee Colony algorithm (ABC) [13], Biogeography-Based Optimization (BBO) [14], Grey Wolf Optimizer (GWO) [15], Whale Optimization Algorithm (WOA) [16], etc. These algorithms have shown great

* Corresponding authors.
E-mail addresses: chenxch8@mail.sysu.edu.cn (X. Chen), isslxc@mail.sysu.edu.cn (X. Liu).

https://doi.org/10.1016/j.asoc.2018.11.047

potential to deal with real-world optimization problems especially in the field of feature selection (FS).

Feature selection (FS) is an important preprocessing step that has a significant impact on the performances of data mining and machine learning techniques [17]. Searching for the optimal feature subset amongst an unabridged dataset is a challenging problem, especially for large-scale datasets. In recent years, many ISOAs are employed to tackle the FS problems. In [18] an efficient Grasshopper Optimization Algorithm (GOA) with evolutionary population dynamics and selection operators has been employed to deal with FS. In [19] an intelligent detection system based on hybridizing GA and random weight network has been proposed to automatically identify the relevant features of the spam emails. In [20] I. Aljarah et al. proposed an asynchronous binary Salp Swarm Algorithm (SSA) with three different updating strategies to tackle the FS problems. In [21] M. Mafarja et al. integrated eight time-varying transfer functions into the Dragonfly Algorithm (DA) to convert the step vector from continuous to a binary space for FS problems.

Grey wolf optimizer is a novel ISOAs by simulating the social hierarchy and hunting behaviour of grey wolves in nature [15]. As GWO has the advantages of simple principle, fewer tuning parameters, fast convergence and strong local search ability, the research on it has made remarkable progress and achieved fruitful rewards. The main applications of GWO include global optimization, machine learning, power engineering, environmental applications, wireless sensor network, medical bioinformatics and image processing [22]. In [23] Emary et al. proposed two novel binary versions of GWO for feature selection by using different updating mechanisms. In [24] GWO was utilized to determine the best combination of control variables to solve optimal reactive power dispatch (ORPD) problem. In [25] a multi-level hybrid clustering routing protocol algorithm based on GWO was proposed for wireless sensor networks. In [26] Song et al. proposed a novel surface wave dispersion curve inversion scheme where GWO was used for parameter estimation in surface waves. In [27] two different approaches for multi-objective binary GWO algorithm were utilized for improving cervix lesion classification in medical bioinformatics. In [28] a novel approach of multilevel thresholding based on GWO was proposed to solve the image segmentation problem.

These applications show that GWO can be widely used to solve a variety of real-world optimization problems. However, similarly to other ISOAs, GWO has its inevitable drawbacks and limitations. The main drawback of GWO is the insufficiency of global search capability as all three alpha, beta, and delta wolves tend to converge to the solution with the same search strategy. The main limitation is that the single search strategy makes GWO not effectively handle various optimization problems with different characteristics [22]. Hence, some modifications have been done to improve the performance of GWO. The modifications can be categorized into four aspects: update mechanisms, new operators, population and social hierarchy, and hybridization. From the aspect of update mechanisms, researchers target on the balance between exploration and exploitation of GWO. To achieve this balance, dynamically tuning the parameters of GWO and adopting different update strategies for search agents are two main means [29–31]. The scheme of adding new operators focuses on improving the performance of GWO by embedding a new operator like crossover operator or a local search algorithm [32–34]. Some researchers investigate the effect of hierarchical structure and do some modifications on it since the hierarchy is a basic and unique feature of GWO [35–37]. As for hybridization, the most common strategy is to integrate GWO with two or more algorithms to have complementary advantages [38,39].

Considering the combination of update mechanisms and new operators, a multi-strategy ensemble GWO (MEGWO) is proposed to boost the efficiency and precision of the original GWO in this paper. The proposed MEGWO incorporates three different search strategies including enhanced global-best lead strategy, adaptable cooperative hunting strategy, and disperse foraging strategy. The enhanced global-best lead strategy can fully explore the excellent experience of leader wolf to enhance the local search ability of GWO. By the adaptable cooperative hunting strategy, the global search capability of the original GWO is enhanced as a result of embedding the one-dimensional update operation. Then, disperse foraging strategy is utilized to further enhance the exploration by relocating a part of search agents to a promising area of the search space. Moreover, the adaptive parameter tuning strategy is used for balancing the exploitation and exploration of the algorithm during the iterative process. The performance of the proposed algorithm is examined on 18 benchmark functions and CEC2014 test set. Non-parametric Wilcoxon test and performance profile are also used to investigate the significance of the results. Furthermore, feature selection is employed to evaluate the effectiveness of MEGWO on real-world applications.

The rest of the paper is organized as follows. In Section 2, we describe the mechanism and mathematical model of GWO. In Section 3, the motivations and details of MEGWO are illustrated. Then the experiments on a wide set of test functions and the application on feature selection of MEGWO are conducted and discussed in Section 4. Finally, conclusions and future direction are given in Section 5.

2. An overview of grey wolf optimizer

GWO is a population-based stochastic algorithm that mimics the leadership hierarchy and hunting strategy of grey wolves in nature [40].

2.1. Leadership hierarchy

Grey wolves in nature have a strict social dominant hierarchy as shown in Fig. 1. According to the order of social status from high to low, the grey wolf is categorized into four levels: alpha (α), beta (β), delta (δ) and omega (ω). Different levels of grey wolves play different roles and bear corresponding responsibilities. The higher the level the wolf belongs to, the better understanding of the location of the prey it has. This mechanism is essential for effective hunting of grey wolves. In order to mathematically model the social hierarchy of wolves when designing GWO, the first three fittest search agents in every iteration are regarded as α, β, and δ. The rest of search agents are treated as ω. During optimization, ω are guided by α, β, and δ towards promising areas of the search space to find the optimal solution.

2.2. Hunting strategy

According to [41], the hunting behaviour of grey wolves mainly consists of three phases: tracking and approaching prey, encircling and harassing prey, and attacking prey. The mathematical model of encircling behaviour is formulated as follows:

$$\boldsymbol{D}^t = \left|\boldsymbol{C}^t \cdot \boldsymbol{X}_p^t - \boldsymbol{X}^t\right| \tag{1}$$

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}_p^t - \boldsymbol{A}^t \cdot \boldsymbol{D}^t \tag{2}$$

where $\boldsymbol{X}_p^t$ and $\boldsymbol{X}^t$ indicate the position vectors of the prey and the wolf at tth iteration, respectively. $\boldsymbol{X}^{t+1}$ stands for the position vector of the wolf at $(t+1)^{th}$ iteration. $\boldsymbol{A}$ and $\boldsymbol{C}$ are coefficient vectors. $\boldsymbol{A} = 2\varphi \cdot \boldsymbol{r}_1 - \varphi$, $\boldsymbol{C} = 2 \cdot \boldsymbol{r}_2$. φ is linearly decreased from 2 to 0 over the course of iterations and $\varphi = 2 - 2 \cdot (t/L)$. L indicates the maximum number of iterations. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are random vectors

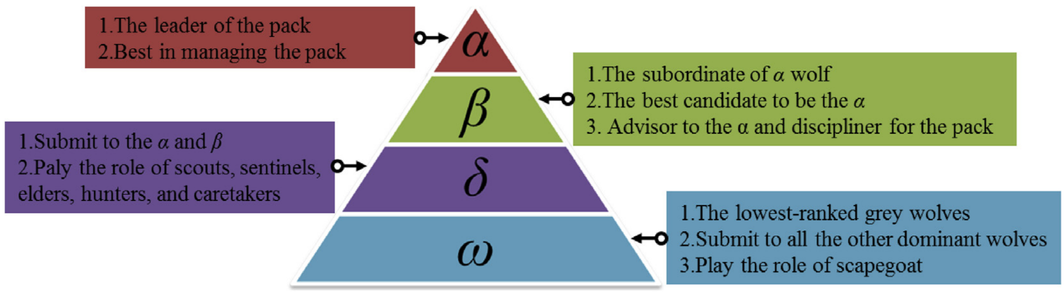


Fig. 1. The leadership hierarchy of wolves and their characteristics.

whose elements are all within [0, 1]. It is noted that the parameters $\boldsymbol{A}$ and $\boldsymbol{C}$ allow wolves to be relocated to any place in the search space around the prey [40].

In the GWO algorithm, it is assumed that α, β, and δ have better knowledge about the potential location of the prey (optimum). Therefore, the first three fitness search agents obtained so far are considered as α, β, and δ, respectively. Then other search agents are assumed as ω and obliged to update their location in the light of α, β, and δ. The mathematical model of hunting behaviour is formulated as follows:

$$\boldsymbol{D}_\alpha^t = \left|\boldsymbol{C}_1^t \cdot \boldsymbol{X}_\alpha^t - \boldsymbol{X}^t\right| \tag{3}$$

$$\boldsymbol{D}_\beta^t = \left|\boldsymbol{C}_2^t \cdot \boldsymbol{X}_\beta^t - \boldsymbol{X}^t\right| \tag{4}$$

$$\boldsymbol{D}_\delta^t = \left|\boldsymbol{C}_3^t \cdot \boldsymbol{X}_\delta^t - \boldsymbol{X}^t\right| \tag{5}$$

$$\boldsymbol{X}_1^t = \boldsymbol{X}_\alpha^t - \boldsymbol{A}_1^t \cdot \boldsymbol{D}_\alpha^t \tag{6}$$

$$\boldsymbol{X}_2^t = \boldsymbol{X}_\beta^t - \boldsymbol{A}_2^t \cdot \boldsymbol{D}_\beta^t \tag{7}$$

$$\boldsymbol{X}_3^t = \boldsymbol{X}_\delta^t - \boldsymbol{A}_3^t \cdot \boldsymbol{D}_\delta^t \tag{8}$$

$$\boldsymbol{X}^{t+1} = \left(\boldsymbol{X}_1^t + \boldsymbol{X}_2^t + \boldsymbol{X}_3^t\right)/3 \tag{9}$$

where $\boldsymbol{X}_\alpha^t$, $\boldsymbol{X}_\beta^t$, and $\boldsymbol{X}_\delta^t$ stand for the positions of α, β, and δ at tth iteration, respectively. $\boldsymbol{D}_\alpha^t$, $\boldsymbol{D}_\beta^t$, and $\boldsymbol{D}_\delta^t$ denote the approximate distances between the current solution and α, β, and δ, respectively. After estimating the approximate distances, the final location of the solution can be calculated by Eq. (9). The pseudo code of original GWO algorithm is presented in Algorithm 1.

3. Multi-strategy ensemble GWO

In GWO, α, β, and δ lead ω towards promising areas to search the optimal solution. This phenomenon can cause premature aggregating in the current optimal position and make GWO fall into the local optimum. Moreover, the complicated optimization problems have various characteristics such as: high dimensionality, nonlinearity, multi-modality, and non-separable [42]. It is difficult for single search strategy of GWO to solve a variety of complex optimization problems. To overcome these issues, we propose MEGWO which incorporates three different search strategies including enhanced global-best lead strategy, adaptable cooperative hunting strategy, and dispersed foraging strategy.

Algorithm 1 The pseudo code of original GWO

Input: Population size N, Maximum number of iterations L
Output: $\boldsymbol{X}_\alpha$ and the best fitness value $f(\boldsymbol{X}_\alpha)$

```
Initialize the search agent population X_i, (i = 1, 2, ..., N)
Initialize φ, A, C, t = 1
Calculate the fitness values of each search agent
X_α = the best search agent
X_β = the second search agent
Xδ = the third search agent
while t < L do
    for i = 1 to N do
        Update the position of search agent X_i^t by Eq.(9)
    end for
    Update φ, A, C
    Evaluate the fitness value of all search agents
    Update X_α^t, X_β^t, X_δ^t
    t = t + 1
end while
```

3.1. Enhanced global-best lead strategy

Inspired by the concept of global-best strategy proposed in global-best harmony search (GHS) [43], an enhanced global-best lead strategy is proposed in this paper. In GHS, the best harmony in the harmony memory is regarded as the global-best candidate solution. GHS modifies the pitch-adjustment step of by randomly selecting one element from the best harmony vector such that the new harmony can mimic the best harmony. The results in [43] indicated that such modifications enhanced the local search ability of HS and allowed it to work efficiently on both continuous and discrete problems. However, this pitch-adjustment approach that the new harmony is selected directly from the best harmony has a bias towards exploitation [44].

The enhanced global-best lead strategy in this paper regards the alpha wolf as the global-best wolf in the pack. The new strategy adopts the same way as GHS to explore the rich experience of the alpha wolf. In addition, it adds a modified differential mutation operation to keep the search from having a bias towards exploitation.

$$\boldsymbol{X}_{i,d}^{t+1} = \begin{cases} \boldsymbol{X}_{\alpha,j_{rand}}^t, & rand_1 < GR \\ \boldsymbol{X}_{\alpha,d}^t + 2\cdot\varphi\cdot r_3\cdot(\boldsymbol{X}_{m,d}^t - \boldsymbol{X}_{n,d}^t), & \text{otherwise} \end{cases} \tag{10}$$

where t indicates the number of current iterations. i, m, and $n \in \{1, 2, \ldots, N\}$, $i \neq m \neq n$ and $d \in \{1, 2, \ldots, D\}$. Herein, N is the population size and D is the number of the dimensions of the solution. j_{rand} is a randomly chosen integer within the range $[1, D]$. φ gradually decreases from 2 to 0 over the course of iterations. $\boldsymbol{X}_\alpha$ stands for the position of α. GR represents the global-best guidance rate which is a user-specified constant between 0 and 1. Both $rand_1$ and r_3 are random numbers between 0 and 1.

It can be seen from Eq. (10) that there are two update operators in the enhanced global-best lead strategy. They are elite decision operator based on the self-cognition of α and elite guidance operator by integrating social cognition into the experience of α. When $rand_1 < GR$, each wolf updates the dth element of its position vector by randomly selecting an element from the position vector of α. This operator can enhance the local search ability and speed up the convergence rate of the algorithm, yet it may reduce the diversity of population and cause the algorithm to trap into local optimum. As a supplement, when $rand_1 \geq GR$, the differential mutation operation which uses $\boldsymbol{X}_\alpha$ as the basis vector is adopted to maintain the individual diversity and expand the scope of the search [45]. The pseudo code of enhanced global-best lead strategy is presented in Algorithm 2.

Algorithm 2 Enhanced global-best lead algorithm

Input: The search agent population $\boldsymbol{X}^t$, $\boldsymbol{X}^t_\alpha$, parameter φ and GR
Output: Updated search agents population $\boldsymbol{X}^{t+1}$,
for $i = 1$ to N **do**
  for $d = 1$ to D **do**
    if $rand_1 < GR$ **then**
      Generate a random integer j_{rand} and $j_{rand} \in \{1, 2, \dots, D\}$
      Update $\boldsymbol{X}^t_{i,d}$ by $\boldsymbol{X}^{t+1}_{i,d} = \boldsymbol{X}^t_{\alpha,j_{rand}}$
    else
      Selected two other search agents $\boldsymbol{X}^t_m$ and $\boldsymbol{X}^t_n$, $m \neq n \neq i$
      Update $\boldsymbol{X}^t_{i,d}$ by $\boldsymbol{X}^{t+1}_{i,d} = \boldsymbol{X}^t_{\alpha,d} + 2 \cdot \varphi \cdot r_3 \cdot (\boldsymbol{X}^t_{m,d} - \boldsymbol{X}^t_{n,d})$
    end if
  end for
end for

3.2. Adaptable cooperative hunting strategy

Like most evolution algorithms, GWO adopts the total-dimensional update operation which changes all elements of the solution vector in each iteration. This mechanism has an excellent performance for optimization of non-separable function. However, it may not be able to achieve a high quality solution for multimodal and separable functions due to the interferences among different dimensions of the solution [46]. ABC [13] is a novel swarm intelligent optimization algorithm, which is inspired by the behaviour of bees finding honey. One-dimensional update operation is an interesting feature of ABC. This operation only changes one element of candidate solution vector in each iteration and utilizes the greedy selection strategy to ensure the new better solution obtained to be reserved for the next iteration [46]. According to the study of [47], the one-dimensional updating operation can provide a higher population diversity than the total-dimensional one and thus makes ABC have a strong global search ability especially for multimodal and separable functions.

In the original GWO, ω wolves encircle and attack prey by teamwork under the leadership of α, β and δ to complete hunting. We regard this update mode as cooperative hunting strategy. Inspired by the one-dimensional update operation of ABC, we integrate the one-dimensional update operation into the framework of original GWO to form an adaptable cooperative hunting strategy. In adaptable cooperative hunting phase, one-dimensional update operation and total-dimensional update operation are alternatively adopted when search agents update their position vectors. Specifically, each search agent will choose one-dimensional update operation with probability SR and total-dimensional one with probability $(1 - SR)$. SR is formulated as follows:

$$SR = SR_{\max} - (SR_{\max} - SR_{\min}) \cdot t/L \tag{11}$$

where SR_{max} and SR_{min} are the upper and lower bounds of SR, respectively.

In this way, the proposed cooperative hunting strategy improves the adaptability of the algorithm on various functions. That is why we name the strategy as adaptable cooperative hunting strategy. The pseudo code of cooperative hunting strategy is presented in Algorithm 3.

Algorithm 3 Adaptable cooperative hunting algorithm

Input: The search agent population $\boldsymbol{X}^t$, parameter φ, A, C and SR
Output: Updated search agents population $\boldsymbol{X}^{t+1}$,
for $i = 1$ to N **do**
  if $rand_2 \leq SR$ **then**
    Generate a random integer j_{rand} and $j_{rand} \in \{1, 2, \dots, D\}$
    $j = j_{rand}$
  else
    $j = 1$ to D
  end if
  for $d = j$ **do**
    Update the position of search agent $\boldsymbol{X}^t_{i,d}$ by Eq.(9)
  end for
end for

3.3. Disperse foraging strategy

Food shortage events may occur in the area where a pack of grey wolves live. This event forces a part of wolves to be dispersed into a new area to forage for surviving. The dispersal behaviour make it more possible that grey wolves migrate to a region with abundant food and improve the viability of wolves in harsh environments. To mathematically model the disperse behaviour, disperse foraging strategy is proposed. In disperse foraging phase, a part of search agents can be repositioned to a promising area of the search space according to the dispersion rate (DR) and update their location as follows [48]:

$$\boldsymbol{X}^{t+1}_{i,d} = \boldsymbol{X}^t_{i,d} + \rho \cdot \boldsymbol{\Delta}^t_{i,d} \cdot \boldsymbol{B}^t_{i,d} \tag{12}$$

$$\boldsymbol{\Delta}^t_{i,d} = (\boldsymbol{X}^t_{m,d} - \boldsymbol{X}^t_{n,d}) \tag{13}$$

where ρ is the scale factor that controls the migration distance of search agents during the foraging phase and $\rho \sim N(0.5, 0.1^2)$. The same setting of ρ can be found in [49]. $\boldsymbol{\Delta}^t_{i,d}$ represents a random migration distance of the search agent. $\boldsymbol{B}^t_{i,d}$ is a logical value which is used to determine whether the search agent is dispersed and can be formulated as follows:

$$\boldsymbol{B}^t_{i,d} = \begin{cases} 1, & rand_3 > DR \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where DR is defined as a linearly decreasing parameter over the iterative course which formulates as follows:

$$DR = DR_{\max} - (DR_{\max} - DR_{\min}) \cdot t/L \tag{15}$$

It can be seen from Eqs. (12) and (15), the scale factor ρ assists search agents to show a more random behaviour throughout optimization process. Meanwhile, the self-adaptive adjustment of DR is utilized to control the scale of search agents which are dispersed to a new area. In the early stage of iterations, a small scale of grey wolves is repositioned in disperse foraging phase with a large DR value. It contributes to provide a fast convergence speed. Then, when the value of DR becomes smaller as the number of iterations increases, search agents are more likely to participate in disperse foraging operation and increase the chance that the algorithm avoids local optimum. The pseudo code of disperse foraging strategy is presented in Algorithm 4.

Algorithm 4 Disperse foraging algorithm

Input: The search agent population $\boldsymbol{X}^t$, parameter DR
Output: Updated search agents population $\boldsymbol{X}^{t+1}$
1: Generate a logic matrix $\boldsymbol{B}^t$ by Eq.(14)
2: Generate the dispersed distance matrix $\boldsymbol{\Delta}^t$ by Eq.(13)
3: Update the position of search agent $\boldsymbol{X}^t$ by Eq.(12)

3.4. Main procedure of MEGWO

The main process of MEGWO is illustrated as Fig. 2. It can be seen from Fig. 2 that the enhanced global-best lead strategy makes search agents fully exploit the search space around the current best solution. The strategy has good local search ability and can help to find a more precise solution. As for adaptable cooperative hunting strategy, one-dimensional update operation is embedded in the framework of the original GWO. It not only retains the advantages of fast convergence speed and strong local search ability of GWO, but also makes full use of the characteristics of one-dimensional update operation to improve the diversity of populations and hence enhance the global search ability. Besides the above two strategies, as a supplementary means, disperse foraging strategy is vital to balance the exploitation and exploration of the algorithm during the iterative course. It can not only guarantee the convergence speed of the algorithm in the early stage of iteration, but also enlarge the search range of the population and prevent the algorithm from falling into local stagnation in the later stage of iteration.

4. Experimental results and discussion

In order to investigate the performance of MEGWO, a great deal of experiments were implemented in this section. 18 classical benchmark functions were used to analyse the impact of the relevant parameters and the effectiveness of the improvement strategies in MEGWO. Furthermore, 30 benchmark functions from CEC2014 [50] were employed to compare MEGWO with other three modified GWO algorithms and seven state-of-the-art ISOAs algorithms. The analysis of computational complexity of the proposed algorithm was conducted as well. Finally, feature selection was applied to validate the performance of MEGWO on real-world problems. All experiments were performed on a PC with Intel(R) Core i7-7700k 4.20 GHz CPU, 8 GB RAM in the Windows10 OS and coded in Matlab R2014b.

4.1. Experimental results on benchmark set1

4.1.1. Benchmark test functions and parameter settings

In this subsection, 18 benchmark test problems with different characteristics were applied to analyse the performance of MEGWO [51,52]. The selected test functions are summarized in Table 1, where C indicates the characteristics of the test functions, which include unimodal (U), multimodal (M), separable (S), and non-separable (N). *Range* denotes the search range of test functions. F_{min} means the global optimal value. The 18 test functions can be classified into three groups: unimodal functions (f_1–f_6), multimodal functions (f_7–f_{12}), shifted functions (f_{13}–f_{18}). Our objective is to obtain the minimum for each function.

For each problem, the population size (N) was 20 and the maximum number of iterations (L) was 2500 for 30-dimensional function and 3500 for 50-dimensional case. The other specific parameters of MEGWO algorithm were set as follows: $GR = 0.8$, $SR_{max} = 1$, $SR_{min} = 0.6$, $DR_{max} = 0.4$ and $DR_{min} = 0$.

4.1.2. The impact of parameter GR

The parameter GR determines how many search agents will adopt elite decision operator based on the self-cognition of α wolf. It is crucial for balance the between exploration and exploitation in the enhanced global-best lead phase. To find appropriate GR value, we performed a large amount of experiments. We varied the value of GR from 0 to 1 with interval length 0.2, and kept the other parameters fixed for all of the test functions with 30 dimensions. Each algorithm independently ran 30 times on each test function while the mean of fitness value (*Mean*) and the standard deviation of fitness value (*Std*) are documented in Table 2. The best results are marked in bold.

It can be seen from Table 2 that the value $GR = 0.8$ provides a best solution compared with other choices of GR values for most of test functions. For functions f_1, f_2, and f_3, when GR is set to 1, the algorithm provides the best solution which even converges to the theoretical optimal one for functions f_1 and f_2. For function f_4, the value of $GR = 0.8$ obtains the best solution. For function f_5, the value $GR = 0.4$, 0.6, 0.8, and 1 provide the same results in terms of the *Mean* and *Std*. For function f_6, all of GR values can help the algorithm to converge to the optimal solution. Since the unimodal functions f_1 to f_6 have only one global optimum, these results show that the exploitation ability of MEGWO can be improved by a large GR value. Functions f_7 to f_{12} are multimodal test functions and appropriate to validate the exploration ability of different algorithms. Based on Table 2, $GR = 0.8$ provides best results compared with other GR values for functions f_7 to f_{12}, which concludes that MEGWO with $GR = 0.8$ obtains best global convergence performance. For shifted functions f_{13} to f_{16}, $GR = 0.8$ also provides very competitive solutions compared to other GR values in terms of *Mean* and *Std*. Therefore, considering the balance between exploration and exploitation in the enhanced global-best lead phase, we conclude that setting GR as 0.8 is an appropriate choice.

4.1.3. The effectiveness of the improvement strategies on MEGWO

As mentioned previously, MEGWO algorithm consists of three main improvement strategies: enhanced global-best lead strategy, adaptable cooperative hunting strategy, and disperse foraging strategy. The objective of this subsection is to validate the effectiveness of these three strategies. To this end, a comparison between MEGWO and its variants were conducted for 18 test functions with 50-dimensions. Herein, its variants are denoted as MEGWO1, MEGWO2 and MEGWO3, respectively. The algorithm adopting the adaptable cooperative hunting strategy and disperse foraging strategy while the enhanced global-best lead strategy is not used is referred to MEGWO1. The algorithm employing enhanced global-best lead strategy, disperse foraging strategy and the adaptable cooperative strategy without one-dimensional update operation is denoted as MEGWO2. The algorithm adopting enhanced global-best lead strategy and adaptable cooperative hunting strategy while disperse foraging strategy is ignored is denoted as MEGWO3. For each test function, 30 independent runs were implemented and L was set to 3500. The *Mean* and *Std* results of GWO, MEGWO1, MEGWO2, MEGWO3 and MEGWO are shown in Table 3. Wilcoxon's signed-rank test at a 5% significant level was employed for evaluated test systems. " $+$ ", " $-$ ", and " $\approx$ " denote that the performance of the proposed MEGWO is superior to, inferior to, or similar to that of corresponding algorithm, respectively [53]. The best results are highlighted in bold.

Observed from Table 3, MEGWO surpasses GWO on 15 test functions and is similar to GWO on 3 test functions. The results show that the performance of GWO has a significant improvement utilizing the combination of three improvement strategy. With respect to MEGWO1, MEGWO shows better performance on 11 test functions. However, it achieves similar performance on 5 test functions f_6, f_9, f_{13}, f_{15}, and f_{16} while it obtains worse results for f_{17} and

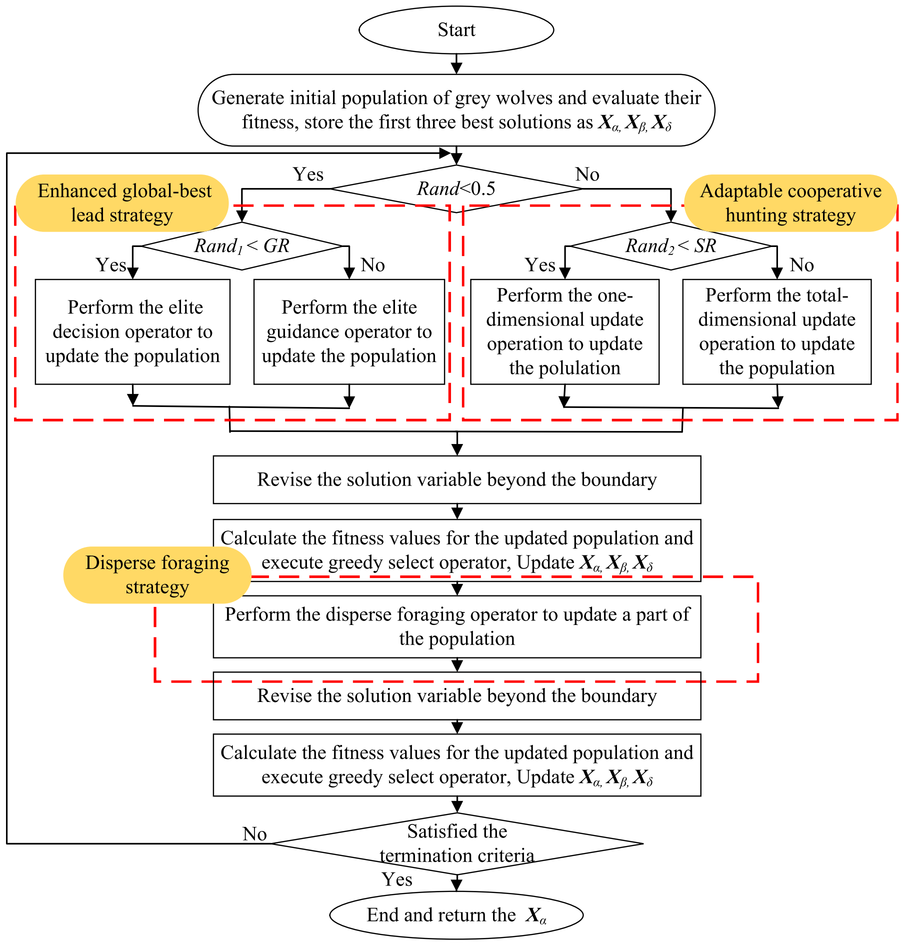


Fig. 2. The main process of MEGWO.

f_{18}. This is due to the fact that the enhanced global-best lead strategy was not used in MEGWO1 which makes the algorithm focus on deep exploration and achieve promising results for multi-modal functions. Compared to MEGWO2, MEGWO obtains better results on 5 functions while it got similar results on 10 functions and worse results on 3 functions f_2, f_3, and f_7. It seems that MEGWO without one-dimensional update operation emphasizes on exploitation and is good at solving unimodal functions problems. With respect to MEGWO3, MEGWO obtains better results on 14 functions and similar results on 4 functions. The performance of MEGWO3 is obviously inferior to MEGWO. Hence, it can be concluded that the disperse foraging strategy plays an important role in improving the efficiency and accuracy of GWO. The convergence characteristics of 5 algorithms on 50-dimensional test function f_1, f_4, f_{11}, and f_{14} are shown in Fig. 3. From Fig. 3, we can observe that MEGWO converges closest to the global optimum for all the functions and with a fastest rate of convergence except for f_1. Similar behaviours can also be observed on most of the rest test functions. Due to the space limit, we omit other resulting figures.

From Table 3 and Fig. 3, it concludes that each update strategy is indispensable and the integration of them is the key to improve the performance of the algorithm efficiently.

4.2. Experiment results and analysis on benchmark set 2

4.2.1. Benchmark functions of CEC2014

In this section, IEEE CEC2014 benchmark functions were utilized [50]. This benchmark set consists of 30 unconstrained optimization problems with different characteristics such as unimodality, multimodality, hybrid and composite modality. More details about the definition of these functions can be found in [50]. According to the guidelines provided by [50], the range of the search space for each variable is [−100, 100]. The termination criteria is defined as be the maximum number of function evaluations (*NFE*), which is $10^4 \times$ *dimension*.

4.2.2. Comparison of MEGWO with other modified GWO algorithms

To validate the superiority of MEGWO, the proposed algorithm were compared with the original GWO [15] and three existing modified GWO algorithms including EPDGWO [32], LGWO [35] and RWGWO [37]. The initial parameters of these algorithms were set to the ones adopted in their original references and are reported in Table 4. *NFE* was set to 3×10^5 for all the algorithms for a fair comparison. For each test function with 30 dimensions, 51 independent runs were implemented. The statistical tests in experiment 4.2 are similar to those of experiment 4.1 while *Mean* and *Std* were used as criterions. The comparison results are presented in Table 5 and the best results are marked in bold. Please note that the results of LGWO and RWGWO are directly taken from their original literatures.

As shown in Table 5, MEGWO beats the original GWO for all of the test functions except for f_{24} and f_{26}. Compared with EPDGWO, MEGWO achieves better results for all of the functions except for f_{24}. With respect to LGWO, MEGWO achieves better results on 17 functions. Meanwhile it gets worse results on 11 functions of the rest ones and similar results on the other 2 functions. Compared with RWGWO, MEGWO could obtain better results on 25 functions and similar results on 1 function. For f_6, f_{24}, f_{29}, and f_{30}, MEGWO is beaten by RWGWO. As Table 5 states, MEGWO shows a significant

Table 1
Test functions used in the experiments.

Name	Function	*C*	*Range*	*Fmin*
Sphere	$f_1(x)=\sum_{i=1}^{n} x_i^2$	US	[−100,100]	0
Schwefel2.22	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	UN	[−10,10]	0
Quartic	$f_3(x)=\sum_{i=1}^{n} ix_i^4+rand(0,1)$	US	[−1.28,1.28]	0
Rosenbrock	$f_4(x)=\sum_{i=1}^{n-1}\left[(x_i-1)^2+100\left(x_{i+1}-x_i^2\right)^2\right]$	UN	[−10,10]	0
Schwefel2.26	$f_5(x)=418.98288727243369*n-\sum_{i=1}^{n} x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	UN	[−500,500]	0
Step	$f_6(x)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	US	[−100,100]	0
Ackley	$f_7(x)=-20\exp(-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)]$ $+20+e$	MN	[−32,32]	0
Rastrigin	$f_8(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	MS	[−5.12,5.12]	0
Griewank	$f_9(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	MN	[−600,600]	0
Levy	$f_{10}(x)=\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+sin^2(3\pi x_1)$ $+\lvert x_n-1\rvert[1+\sin^2(3\pi x_n)]$	MN	[−10,10]	0
Penalized1	$f_{11}(x)=\frac{\pi}{n}\{10\sin^2(\pi y_i)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+$ $(y_n-1)^2\}+\sum_{i=1}^{n}u(x_i,10,100,4),\ y_i=1+\frac{1}{4}(x_i+1),$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\ 0\qquad\quad, -a\le x\le a\\ k(-x_i-a)^m, x_i<-a\end{cases}$	MN	[−50,50]	0
Penalized2	$f_{12}(x)=\frac{1}{10}\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2\left[1+\sin^2(3\pi x_{i+1})\right]\right.$ $\left.+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	MN	[−50,50]	0
Shifted Sphere	$f_{13}(z)=\sum_{i=1}^{n}z_i^2-450,\ z=x-o$	US	[−100,100]	−450
Shifted Schwefel2.21	$f_{14}(z)=\max_{i=1}^{n}\{\lvert z_i\rvert\}-450,\ z=x-o$	UN	[−100,100]	−450
Shifted Griewank	$f_{15}(z)=1+\sum_{i=1}^{n}\frac{z_i^2}{4000}-\prod_{i=1}^{n}\cos(\frac{z_i}{\sqrt{i}})-180,\ z=x-o$	MN	[−600,600]	−180
Shifted Ackley	$f_{16}(z)=-20\exp[-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}z_i^2}]-\exp[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)]$ $+20+e-140,\ z=x-o$	MN	[−32,32]	−140
Shifted Rastrigin	$f_{17}(z)=\sum_{i=1}^{n}(z_i^2-10\cos(2\pi z_i)+10)-330,\ z=x-o$	MS	[−5,12,5.12]	−330
Shifted Rosenbrock	$f_{18}(z)=\sum_{i=1}^{n-1}\left[(z_i-1)^2+100\left(z_{i+1}-z_i^2\right)^2\right]+390,$ $z=x-o+1$	UN	[−100,100]	390

improvement over GWO, EPDGWO and RWGWO while LGWO also provides a very competitive result compared to MEGWO. To make our superiority more distinguished, we introduce performance profiles [62] as a tool for evaluation of all relevant algorithms on the CEC2014 test sets.

Performance profiles can clearly reflect the performance of one algorithm in the algorithms set G which consists of n_g algorithms on a functions set F which consists of n_f test functions. The mean fitness value is chosen as the performance metric. $\mu_{f,g}$ is defined as the mean fitness value after performing algorithm g on function f. If we compare the performance on function f by algorithm g with the best performance by the algorithms in set G on this function, the performance ratio $r_{f,g}$ is calculated as follows:

$$r_{f,g}=\frac{\mu_{f,g}}{\min\{\mu_{f,g}:g\in G\}} \tag{16}$$

To obtain an overall assessment of the performance of algorithm g on all functions, we define

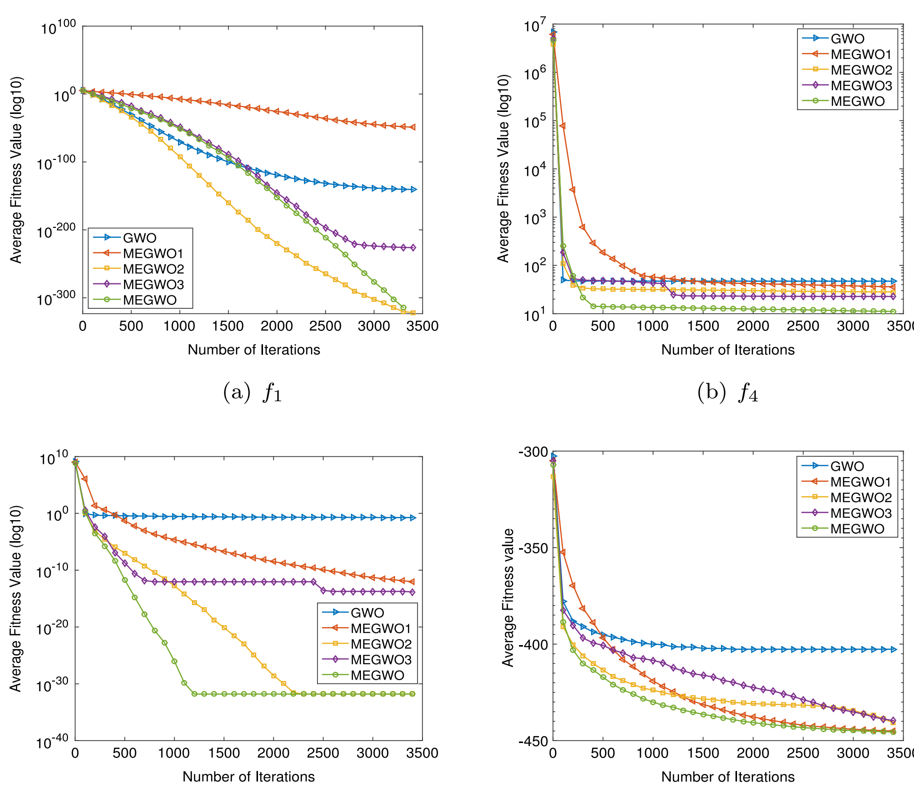
$$\rho_g(\tau)=\frac{1}{n_f}size\{f\in F:r_{f,g}\le\tau\} \tag{17}$$

which is the probability for algorithm $g\in G$ that a performance ratio $r_{f,g}$ is within a factor $\tau\in\mathcal{R}$ of the best possible ratio [62].

Fig. 4 presents the performance ratios of the mean fitness value for the five algorithms on the CEC2014 test set. The results are displayed by a log scale2. From Fig. 4, it is clear that MEGWO has the highest probability to be the optimal algorithm and the probability that MEGWO is the winner on a given test function is about 0.57 when $\tau=0$. When $\tau=1$ both LGWO and MEGWO are the winner on approximately 76% of the problems. Though the MEGWO obtains similar performance to LGWO when τ is in the interval [9, 18], the performance curve of MEGWO lies above all the other curves and MEGWO can achieve optimality for about 97% problems when $\tau\ge 18$. From all above comparisons, it concludes

Table 2
Experimental results of MEGWO using different *GR* values for 18 test functions ($D = 30$).

Fun		$GR = 0$	$GR = 0.2$	$GR = 0.4$	$GR = 0.6$	$GR = 0.8$	$GR = 1$
f_1	Mean	2.08E−59	1.42E−132	1.18E−174	1.13E−215	4.30E−296	**0.00E+00**
	Std	1.51E−59	2.45E−132	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**
f_2	Mean	2.65E−36	8.91E−82	6.01E−106	3.84E−135	2.14E−193	**0.00E+00**
	Std	3.28E−36	1.39E−81	5.54E−106	6.23E−135	0.00E+00	**0.00E+00**
f_3	Mean	4.95E−03	2.51E−03	1.84E−03	8.17E−04	6.20E−04	**4.52E−04**
	Std	3.98E−03	8.59E−04	1.13E−03	4.80E−04	1.58E−04	**1.12E−04**
f_4	Mean	1.57E+01	1.63E+01	1.62E+01	1.56E+01	**1.01E+01**	1.02E+01
	Std	1.34E+00	4.94E−01	5.67E−01	6.38E−01	**2.80E−01**	8.81E−00
f_5	Mean	1.97E+02	−6.06E−12	**−1.82E−12**	**−1.82E−12**	**−1.82E−12**	**−1.82E−12**
	Std	1.37E+02	1.05E−12	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_6	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	Mean	1.10E−14	5.03E−15	2.73E−15	2.73E−15	**2.66E−15**	1.45E−14
	Std	4.10E−15	2.05E−15	0.00E+00	2.05E−15	**0.00E+00**	1.03E−14
f_8	Mean	6.63E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	5.74E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{10}	Mean	5.73E−17	3.77E−30	**1.50E−32**	**1.50E−32**	**1.50E−32**	2.01E−15
	Std	5.54E−17	3.38E−30	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.47E−15
f_{11}	Mean	6.36E−19	1.70E−32	**1.57E−32**	**1.57E−32**	**1.57E−32**	1.41E−20
	Std	4.28E−19	1.29E−33	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.44E−20
f_{12}	Mean	2.85E−17	7.97E−29	**1.35E−32**	**1.35E−32**	**1.35E−32**	1.46E−12
	Std	3.07E−17	1.37E−28	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.52E−12
f_{13}	Mean	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	**−4.50E+02**	−4.50E+02
	Std	1.33E−08	1.64E−09	2.02E−08	2.68E−08	**1.13E−09**	6.61E−09
f_{14}	Mean	−4.49E+02	−4.49E+02	−4.49E+02	−4.49E+02	**−4.50E+02**	−4.49E+02
	Std	5.54E−02	3.29E−01	4.65E−01	2.93E−01	**2.89E−02**	1.92E−01
f_{15}	Mean	−1.80E+02	−1.80E+02	−1.80E+02	−1.80E+02	**−1.80E+02**	−1.80E+02
	Std	4.27E−03	1.66E−05	1.12E−05	9.54E−05	**2.19E−06**	2.62E−05
f_{16}	Mean	−1.39E+02	−1.40E+02	−1.40E+02	−1.40E+02	**−1.40E+02**	−1.40E+02
	Std	1.01E−04	7.25E−06	1.44E−05	4.44E−05	**4.36E−06**	5.56E−06
f_{17}	Mean	−3.29E+02	−3.29E+02	−3.30E+02	−3.30E+02	**−3.30E+02**	−3.29E+02
	Std	5.74E−01	1.15E+00	2.47E−02	2.54E−01	**5.76E−03**	1.15E+00
f_{18}	Mean	4.65E+02	4.68E+02	4.65E+02	4.74E+02	**4.49E+02**	4.80E+02
	Std	3.85E+01	3.38E+01	3.32E+01	3.67E+01	**2.14E+01**	4.94E+01



(a) f_1 (b) f_4 (c) f_{11} (d) f_{14}

Fig. 3. Convergence curves on 50-dimensional test functions f_1, f_4, f_{11}, and f_{14} of 5 algorithms.

Table 3
Experimental results of GWO, MEGWO1, MEGWO2, MEGWO3, MEGWO on 18 functions ($D = 50$).

Fun	Index	GWO		MEGWO1		MEGWO2		MEGWO3		MEGWO
f_1	Mean	2.62E−141	+	6.50E−50	+	2.22E−322	+	6.26E−227	+	**0.00E+00**
	Std	2.70E−141		1.04E−49		0.00E+00		0.00E+00		**0.00E+00**
f_2	Mean	1.21E−82	+	3.43E−30	+	**9.83E−280**	−	1.62E−195	+	9.72E−245
	Std	9.16E−83		6.02E−30		**0.00E+00**		0.00E+00		0.00E+00
f_3	Mean	6.11E−04	+	4.60E−03	+	**4.18E−04**	−	1.30E−03	+	5.13E−04
	Std	3.96E−04		1.20E−03		**2.36E−04**		7.84E−04		2.93E−04
f_4	Mean	4.71E+01	+	3.56E+01	+	2.80E+01	+	2.27E+01	+	**1.11E+01**
	Std	9.68E+01		4.08E+01		1.93E+01		2.39E+01		**1.79E+01**
f_5	Mean	1.19E+04	+	3.55E+01	+	**1.46E−11**	≈	2.30E−10	+	**1.46E−11**
	Std	6.98E+02		5.72E+01		**0.00E+00**		6.11E−10		**0.00E+00**
f_6	Mean	**0.00E+00**	≈	**0.00E+00**	≈	**0.00E+00**	≈	**0.00E+00**	≈	**0.00E+00**
	Std	**0.00E+00**		**0.00E+00**		**0.00E+00**		**0.00E+00**		**0.00E+00**
f_7	Mean	1.12E−14	+	2.90E−14	+	**3.02E−15**	−	9.77E−15	+	4.09E−15
	Std	3.00E−15		5.35E−15		**1.12E−15**		5.80E−15		1.83E−15
f_8	Mean	**0.00E+00**	≈	3.55E−16	+	**0.00E+00**	≈	**0.00E+00**	≈	**0.00E+00**
	Std	**0.00E+00**		1.12E−15		**0.00E+00**		**0.00E+00**		**0.00E+00**
f_9	Mean	**0.00E+00**	≈	**0.00E+00**	≈	**0.00E+00**	≈	**0.00E+00**	≈	**0.00E+00**
	Std	**0.00E+00**		**0.00E+00**		**0.00E+00**		**0.00E+00**		**0.00E+00**
f_{10}	Mean	5.36E+00	+	3.61E−11	+	**1.50E−32**	≈	3.63E−01	+	**1.50E−32**
	Std	9.97E−01		2.71E−11		**0.00E+00**		8.03E−01		**0.00E+00**
f_{11}	Mean	1.79E−01	+	6.07E−13	+	**1.57E−32**	≈	5.25E−15	+	**1.57E−32**
	Std	5.80E−02		3.46E−13		**2.89E−48**		1.34E−14		**2.89E−48**
f_{12}	Mean	1.90E+00	+	1.18E−11	+	**1.35E−32**	≈	7.06E−20	+	**1.35E−32**
	Std	4.55E−01		9.77E−12		**2.89E−48**		2.23E−19		**2.89E−48**
f_{13}	Mean	1.15E+04	+	**−4.50E+02**	≈	**−4.50E+02**	≈	−4.49E+02	+	**−4.50E+02**
	Std	5.48E+03		5.29E−08		4.01E−07		2.85E−01		**1.75E−09**
f_{14}	Mean	−4.03E+02	+	−4.45E+02	+	−4.42E+02	+	−4.40E+02	+	**−4.46E+02**
	Std	5.04E+00		1.49E+00		3.13E+00		1.33E+00		**1.01E+00**
f_{15}	Mean	−6.39E+01	+	**−1.80E+02**	≈	**−1.80E+02**	≈	−1.79E+02	+	**−1.80E+02**
	Std	5.07E+01		3.18E−05		1.90E−03		7.10E−02		**2.88E−06**
f_{16}	Mean	−1.26E+02	+	**−1.40E+02**	≈	**−1.40E+02**	≈	**−1.40E+02**	≈	**−1.40E+02**
	Std	2.11E+00		8.62E−05		5.18E−01		6.64E−02		**1.99E−06**
f_{17}	Mean	−7.44E+01	+	**−3.30E+02**	−	−2.30E+02	+	−3.28E+02	+	−3.29E+02
	Std	2.99E+01		**4.81E−01**		1.47E+01		8.42E−01		8.48E−01
f_{18}	Mean	1.10E+09	+	**5.07E+02**	−	5.64E+02	+	1.18E+03	+	5.26E+02
	Std	8.26E+08		**3.71E+01**		9.16E+01		1.20E+02		4.19E+01
+/−/≈		15/0/3		11/2/5		5/3/10		14/0/4		

Table 4
The parameter settings of algorithms.

Algorithm	Year	Literature	Parameter settings
In Section 4.2			
MEGWO			$N = 20$, $MaxDT = 7500$, $\varphi_0 = 2$, $SR_{max} = 1$, $SR_{min} = 0.6$, $RW_{max} = 0.4$, $RW_{min} = 0$, $GR = 0.8$
GWO	2014	[15]	$N = 40$, $MaxDT = 7500$, $a_0 = 2$
EPDGWO	2014	[32]	$N = 40$, $MaxDT = 7500$, $a_0 = 2$
LGWO	2017	[35]	$N = 40$, $MaxDT = 7500$, $a_0 = 2$, $\beta \sim U(0, 2)$, $p \sim U(0, 1)$
RWGWO	2018	[37]	$N = 40$, $MaxDT = 7500$, $a_0 = 2$
DE/BBO	2011	[54]	$N = 100$, $MaxDT = 3000$, $CR = 0.9$
MEABC	2014	[55]	$N = 40$, $MaxDT = 7500$, $C = 1.5$
ACS	2015	[56]	$N = 40$, $MaxDT = 7500$, $\alpha = 1$, $\beta = 10$, $\gamma = 1.5$, $pa = 0.25$
SinDE	2015	[57]	$N = 40$, $MaxDT = 7500$, $freq = 0.25$
WOA	2016	[16]	$N = 40$, $MaxDT = 7500$, $l \sim U(-1, 1)$, $p \sim U(0, 1)$
BHS	2016	[58]	$N = 40$, $MaxDT = 7500$, $BW_{max} = 0.1$, $BW_{min} = 0.0001$, $PAR_{max} = 0.99$, $PAR_{min} = 0.35$, $HMCR = 0.95$
CLPSO-LOT	2018	[59]	$N = 40$, $MaxDT = 7500$, $w_{max} = 0.9$, $w_{min} = 0.4$, $C_1 = C_2 = 1.49445$, $m = 7$, $a = 0.2$, $g = 30$
In Section 4.4			
MEGWO			$N = 5$, $MaxDT = 100$, $\varphi_0 = 2$, $SR_{max} = 1$, $SR_{min} = 0.6$, $RW_{max} = 0.4$, $RW_{min} = 0$, $GR = 0.8$
SinDE	2015	[57]	$N = 10$, $MaxDT = 100$, $freq = 0.25$
SRPSO	2015	[60]	$N = 10$, $MaxDT = 100$, $\omega_i = 1.05$, $\omega_f = 0.5$, $C_1 = C_2 = 1.49$
WOASA	2017	[61]	$N = 10$, $MaxDT = 100$, $l \sim U(-1, 1)$, $p \sim U(0, 1)$, $T = 20$

that MEGWO performs significantly better than that other GWO variants.

4.2.3. Comparison of MEGWO with other state-of-the-art algorithms

To further show the superiority of MEGWO, the performance of MEGWO was compared with seven state-of-the-art optimizers, such as ACS [56], MEABC [55], SinDE [57], BHS [58], DE/BBO [54], WOA [16] and CLPSO-LOT [59] for CEC2014 problems by the mean of fitness value and standard deviation of the solutions of objective functions. The parameter settings of all algorithms in this subsection can be found in Table 4. 51 independent runs were carried out for all test functions with 30 dimensions. Table 6 presents the experimental results obtained by MEGWO and other seven algorithms. The best results are marked in bold. Please note that the results of BHS and DE/BBO are directly taken from [52].

Table 5
Experimental results of GWO, EPDGWO, LGWO, RWGWO, and MEGWO for CEC2014 ($D = 30$).

Fun	Index	GWO		EPDGWO		LGWO		RWGWO		MEGWO
f_1	Mean	3.32E+07	+	7.91E+07	+	**5.24E+00**	−	8.02E+06	+	1.50E+06
	Std	2.02E+07		2.05E+07		**3.85E+00**		3.31E+06		3.13E+05
f_2	Mean	1.01E+09	+	4.54E+09	+	**0.00E+00**	≈	2.23E+05	+	**0.00E+00**
	Std	1.15E+09		7.03E+08		**0.00E+00**		5.51E+05		**0.00E+00**
f_3	Mean	2.85E+04	+	1.74E+04	+	1.41E+ 01	+	3.16E+02	+	**0.00E+00**
	Std	6.80E+03		3.77E+03		2.24E−02		4.34E+02		**0.00E+00**
f_4	Mean	1.94E+02	+	3.53E+02	+	3.46E+01	+	3.41E+01	+	**1.46E+01**
	Std	5.82E+01		4.78E+01		1.54E+01		1.80E+01		**1.06E+01**
f_5	Mean	2.10E+01	+	2.09E+01	+	2.06E+01	+	2.05E+01	+	**2.02E+01**
	Std	4.83E+ 02		4.62E+ 02		3.11E+ 02		7.46E+ 02		**2.20E+ 02**
f_6	Mean	1.16E+01	+	2.42E+01	+	**9.72E+00**	−	9.84E+00	−	1.12E+01
	Std	2.64E+00		1.81E+00		**3.41E+00**		3.49E+00		4.30E+00
f_7	Mean	7.67E+00	+	4.19E+01	+	**0.00E+00**	−	2.53E−01	+	4.88E−05
	Std	4.64E+00		7.57E+00		**0.00E+00**		1.43E−01		8.06E−05
f_8	Mean	6.50E+01	+	1.64E+02	+	2.90E+01	+	4.38E+01	+	**0.00E+00**
	Std	1.45E+01		1.79E+01		2.64E+01		8.48E+00		**0.00E+00**
f_9	Mean	8.54E+01	+	2.18E+02	+	8.14E+01	+	6.33E+01	+	**5.90E+01**
	Std	3.30E+01		1.70E+01		3.16E−01		1.30E+01		**8.59E−02**
f_{10}	Mean	1.80E+03	+	5.65E+03	+	2.00E+03	+	9.61E+02	+	**3.25E+00**
	Std	4.93E+02		6.35E+02		4.52E+02		2.72E+02		**1.16E+00**
f_{11}	Mean	2.90E+03	+	6.88E+03	+	2.00E+03	+	2.68E+03	+	**1.70E+03**
	Std	7.24E+02		3.58E+02		6.41E+00		3.68E+02		**2.94E+00**
f_{12}	Mean	2.12E+00	+	2.53E+00	+	**9.28E−02**	−	5.45E−01	+	2.36E−01
	Std	9.58E−01		2.86E−01		**2.98E−01**		1.66E−01		4.50E−01
f_{13}	Mean	3.74E−01	+	9.10E−01	+	3.89E−01	+	2.80E−01	+	**2.39E−01**
	Std	8.88E−02		1.38E−01		2.44E−02		6.30E−02		**1.68E−02**
f_{14}	Mean	7.49E−01	+	1.14E+01	+	4.29E−01	+	4.23E−01	+	**2.32E−01**
	Std	1.40E+00		2.94E+00		3.72E−02		2.15E−01		**2.14E−02**
f_{15}	Mean	2.06E+01	+	8.43E+01	+	7.52E+00	+	8.81E+00	+	**5.76E+00**
	Std	2.20E+01		5.29E+01		4.09E+00		1.51E+00		**6.04E−01**
f_{16}	Mean	1.09E+01	+	1.26E+01	+	1.06E+01	+	1.03E+01	+	**9.15E+00**
	Std	5.80E−01		3.15E−01		2.14E−02		6.11E−01		**2.01E−02**
f_{17}	Mean	6.28E+05	+	2.40E+06	+	**1.64E+03**	−	5.71E+05	+	2.57E+04
	Std	6.11E+05		1.13E+06		**6.99E+02**		4.10E+05		1.29E+04
f_{18}	Mean	5.27E+06	+	4.64E+07	+	1.14E+03	+	6.52E+03	+	**4.49E+01**
	Std	1.34E+07		2.22E+07		2.90E+01		4.62E+03		**1.10E+01**
f_{19}	Mean	2.56E+01	+	3.54E+01	+	9.84E+00	+	1.14E+01	+	**5.82E+00**
	Std	1.77E+01		5.31E+00		1.99E−02		2.03E+00		**1.33E−02**
f_{20}	Mean	1.31E+04	+	4.00E+03	+	8.17E+01	+	6.27E+02	+	**3.27E+01**
	Std	5.26E+03		1.92E+03		4.71E+01		1.12E+03		**9.40E+00**
f_{21}	Mean	4.97E+05	+	7.31E+05	+	1.54E+03	+	2.58E+05	+	**8.28E+02**
	Std	1.05E+06		3.74E+05		2.07E+03		1.76E+05		**2.76E+02**
f_{22}	Mean	2.50E+02	+	5.59E+02	+	2.42E+02	+	2.08E+02	+	**1.83E+02**
	Std	1.16E+02		1.36E+02		2.37E+01		1.29E+02		**1.53E+01**
f_{23}	Mean	3.28E+02	+	3.35E+02	+	**2.00E+02**	−	3.15E+02	+	3.15E+02
	Std	4.16E+00		4.73E+00		**2.86E−01**		2.77E−01		3.96E−01
f_{24}	Mean	2.00E+02	−	2.00E+02	−	**2.00E+02**	−	2.00E+02	−	2.17E+02
	Std	7.27E−04		1.45E−03		**2.18E−05**		3.04E−03		6.46E−01
f_{25}	Mean	2.11E+02	+	2.12E+02	+	**2.00E+02**	−	2.04E+02	+	2.01E+02
	Std	2.04E+00		4.77E+00		**3.98E−03**		1.18E+00		7.96E−01
f_{26}	Mean	1.00E+02	≈	1.01E+02	+	1.00E+02	≈	1.00E+02	≈	**1.00E+02**
	Std	9.62E−02		1.28E−01		4.00E−02		7.36E−02		**2.61E−02**
f_{27}	Mean	4.33E+02	+	8.73E+02	+	**3.99E+02**	−	4.09E+02	+	4.06E+02
	Std	1.82E+01		1.66E+02		**1.00E+00**		6.09E+00		1.85E+00
f_{28}	Mean	9.14E+02	+	1.21E+03	+	**2.04E+02**	−	4.34E+02	+	8.88E+02
	Std	6.63E+01		1.43E+02		**1.86E−02**		8.45E+00		3.32E+01
f_{29}	Mean	2.90E+05	+	1.35E+06	+	2.47E+02	−	**2.14E+02**	−	9.37E+02
	Std	1.57E+06		2.53E+06		2.19E+01		**2.37E+00**		4.67E+00
f_{30}	Mean	2.98E+04	+	4.21E+04	+	3.02E+04	+	**6.69E+02**	−	1.36E+03
	Std	1.57E+04		1.38E+04		5.19E+02		**2.14E+02**		2.48E+02
+/−/≈		28/1/1		29/1/0		17 /11 /2		25 /4 /1		

To intuitively verify the significant difference between MEGWO and other seven state-of-the-art algorithms, the paired Wilcoxon signed-rank tests were performed with a level of significance $\alpha = 0.05$ in this subsection [53]. The results of Wilcoxon signed-rank tests of MEGWO and other seven comparison algorithms on 30-dimensional functions from CEC2014 test set are listed in Table 7, where R^+ indicates the sum of ranks for the problems in which MEGWO outperforms those algorithms for comparison and R^- denotes the sum of ranks on the opposite. p-value represents something about how significant differences the result is: the smaller the p-value, the stronger the evidence. n represents the total number of test functions while w, t, and l indicate the number of MEGWO superior, inferior, or similar to the corresponding algorithm, respectively [63]. As Table 7 states, the number of functions on which MEGWO performs better than ACS, MEABC, SinDE, BHS, DE/BBO, WOA and CLPSO-LOT are 28, 26, 26, 28, 24, 29, and 21, respectively. MEGWO shows a significant improvement over ACS, BHS, and WOA with a level of significance $\alpha = 0.001$, over MEABC, SinDE, DE/BBO and CLPSO-LOT with $\alpha = 0.01$. It concludes that the proposed algorithm MEGWO provides very competitive results compared to other seven state-of-the-art algorithms.

It is well-known that the ability to well-balance exploration and exploitation is essential for algorithms targeting on optimization problems with different characteristics. Firstly, compared with

Table 6
Experimental results of MEGWO and other seven state-of-the-art algorithms for CEC2014 ($D = 30$).

Fun	Index	ACS	MEABC	SinDE	BHS	DE/BBO	WOA	CLPSO-LOT	MEGWO
f_1	Mean	2.24E+06	9.09E+06	6.27E+06	9.33E+06	9.80E+06	1.44E+07	**3.12E+05**	1.50E+06
	Std	6.14E+05	3.38E+06	3.78E+06	6.93E+06	2.66E+06	6.73E+06	**7.07E+04**	3.13E+05
f_2	Mean	1.00E+10	1.93E+02	4.98E+01	1.09E+04	3.69E+01	1.33E+05	1.68E+02	**0.00E+00**
	Std	0.00E+00	3.75E+02	2.55E+02	8.76E+03	1.11E+02	4.26E+04	2.61E+02	**0.00E+00**
f_3	Mean	2.19E−04	9.73E+02	5.27E+00	1.26E+04	5.86E−01	1.71E+04	5.14E+01	**0.00E+00**
	Std	1.20E−04	7.45E+02	2.48E+01	1.05E+04	1.08E+00	1.42E+04	7.24E+01	**0.00E+00**
f_4	Mean	3.98E+01	1.92E+01	**4.82E+00**	1.16E+02	6.21E+01	1.58E+02	6.07E+01	1.46E+01
	Std	3.41E+01	2.56E+01	**1.02E+01**	3.28E+01	1.91E+01	4.04E+01	3.08E+01	1.06E+01
f_5	Mean	2.02E+01	2.02E+01	2.09E+01	2.06E+01	2.04E+01	2.03E+01	2.03E+01	**2.02E+01**
	Std	8.17E−02	2.52E−02	5.56E−02	3.04E−01	3.82E−02	1.92E−01	1.49E−01	**2.20E−02**
f_6	Mean	2.48E+01	1.39E+01	**8.81E−02**	1.48E+01	6.32E+00	3.39E+01	8.00E+00	1.12E+01
	Std	2.21E+00	1.47E+00	**2.63E−01**	2.31E+00	6.03E+00	3.64E+00	1.09E+00	1.30E+00
f_7	Mean	8.75E−06	3.90E−06	1.16E−03	2.48E−02	5.13E−14	4.51E−01	**1.55E−16**	4.88E−05
	Std	2.23E−05	1.25E−05	2.97E−03	4.37E−02	5.71E−14	1.10E−01	**7.93E−17**	8.06E−05
f_8	Mean	5.61E+01	**0.00E+00**	2.22E+01	4.26E−10	4.46E−14	1.78E+02	5.71E+01	**0.00E+00**
	Std	1.01E+01	**0.00E+00**	1.96E+01	4.03E−11	5.61E−14	2.67E+01	1.22E+01	**0.00E+00**
f_9	Mean	1.08E+02	**5.20E+01**	1.54E+02	6.14E+01	6.23E+01	2.35E+02	6.75E+01	5.90E+01
	Std	1.82E+01	**8.32E−02**	1.40E+01	1.36E+01	6.77E+00	5.17E+01	1.47E+01	8.59E−02
f_{10}	Mean	2.47E+03	6.00E−01	4.88E+02	**1.01E−01**	1.27E+01	3.77E+03	4.69E+02	3.25E+00
	Std	4.48E+02	7.77E−01	4.57E+02	**4.24E−02**	3.11E+00	8.32E+02	1.95E+02	1.16E+00
f_{11}	Mean	3.59E+03	1.78E+03	6.26E+03	1.91E+03	3.07E+03	4.63E+03	**2.16E+02**	1.70E+03
	Std	3.64E+02	2.39E+02	3.16E+02	4.15E+02	3.01E+00	9.85E+02	**2.23E+00**	2.94E+00
f_{12}	Mean	1.00E+00	1.59E−01	1.95E+00	**1.22E−01**	5.98E−01	1.54E+00	1.51E−01	2.36E−01
	Std	1.25E−01	2.53E−02	2.61E−01	**1.98E−02**	7.57E−02	4.70E−01	4.08E−02	4.50E−01
f_{13}	Mean	3.26E−01	2.66E−01	2.85E−01	3.95E−01	3.18E−01	5.21E−01	3.24E−01	**2.39E−01**
	Std	4.53E−02	3.88E−02	3.86E−02	8.61E−02	3.36E−02	9.74E−02	6.41E−02	**1.68E−02**
f_{14}	Mean	2.46E−01	2.32E−01	2.50E−01	2.88E−01	2.59E−01	2.61E−01	**2.17E−01**	2.32E−01
	Std	3.06E−02	2.69E−02	4.06E−02	5.45E−02	2.21E−02	6.14E−02	**2.08E−02**	2.14E−02
f_{15}	Mean	1.33E+01	6.48E+00	1.49E+01	2.45E+01	7.50E+00	6.45E+01	**4.26E+00**	5.76E+00
	Std	1.95E+00	8.37E−01	7.86E−01	1.33E+01	8.46E−01	2.28E+01	**1.36E−01**	6.04E−01
f_{16}	Mean	1.25E+01	9.84E+00	1.20E+01	9.15E+00	1.02E+01	1.23E+01	1.10E+01	**9.15E+00**
	Std	2.68E−01	4.32E−01	2.56E−01	8.2E−01	3.21E−01	5.53E−01	7.82E−01	**2.01E−01**
f_{17}	Mean	**4.10E+03**	3.11E+06	1.59E+05	1.44E+06	7.26E+05	2.06E+06	3.01E+04	2.57E+04
	Std	**5.98E+02**	1.38E+06	1.28E+05	1.05E+06	3.18E+05	1.60E+06	2.06E+04	1.29E+04
f_{18}	Mean	1.56E+02	1.63E+03	4.85E+02	1.81E+03	1.30E+03	4.58E+03	2.90E+03	**4.49E+01**
	Std	2.12E+01	1.72E+03	8.04E+02	2.08E+03	1.31E+03	5.42E+03	2.47E+03	**1.10E+01**
f_{19}	Mean	7.92E+00	6.78E+00	6.28E+00	1.60E+01	**4.73E+00**	4.34E+01	7.94E+00	5.82E+00
	Std	6.51E−01	7.45E−01	8.21E+00	2.63E+01	**3.84E−01**	3.56E+01	1.03E+00	8.33E−01
f_{20}	Mean	6.27E+01	9.04E+03	1.30E+02	1.29E+04	9.82E+02	8.14E+03	2.38E+02	**3.27E+01**
	Std	9.87E+00	5.77E+03	1.18E+02	8.13E+03	3.95E+02	5.98E+03	4.84E+01	**9.40E+00**
f_{21}	Mean	1.36E+03	3.02E+05	1.12E+04	2.94E+05	1.20E+05	1.06E+06	5.54E+04	**8.28E+02**
	Std	3.04E+02	1.20E+05	1.17E+04	2.26E+05	6.11E+04	6.84E+05	3.37E+04	**2.76E+02**
f_{22}	Mean	2.70E+02	3.06E+02	**1.29E+02**	5.36E+02	8.84E+02	7.68E+02	2.35E+02	1.83E+02
	Std	9.04E+01	9.29E+01	**7.05E+01**	1.83E+02	2.89E+01	2.87E+02	9.61E+01	9.53E+01
f_{23}	Mean	3.15E+02	3.15E+02	3.15E+02	3.16E+02	3.15E+02	3.25E+02	3.15E+02	**3.15E+02**
	Std	8.38E−13	2.47E−01	6.21E−03	5.32E−01	4.02E−13	6.46E+00	8.77E−01	**3.96E−13**
f_{24}	Mean	2.25E+02	2.27E+02	2.25E+02	2.32E+02	2.23E+02	2.06E+02	**2.00E+02**	2.17E+02
	Std	4.91E+00	3.43E+00	3.01E+00	5.30E+00	9.81E−01	3.79E−01	**1.12E−03**	6.46E−01
f_{25}	Mean	2.07E+02	2.08E+02	2.07E+02	2.14E+02	2.07E+02	2.23E+02	2.07E+02	**2.01E+02**
	Std	1.20E+00	1.29E+00	1.65E+00	4.91E+00	9.52E−01	1.60E+01	1.59E+00	**7.96E−01**
f_{26}	Mean	1.00E+02	1.00E+02	1.04E+02	1.73E+02	1.00E+02	1.00E+02	1.00E+02	**1.00E+02**
	Std	5.24E−02	7.35E−02	1.97E+01	4.50E+01	3.69E−02	1.26E−01	5.13E−02	**2.61E−02**
f_{27}	Mean	4.07E+02	4.11E+02	**3.07E+02**	7.30E+02	9.10E+02	1.16E+03	4.86E+02	4.06E+02
	Std	2.08E+00	4.42E+00	**1.15E+00**	1.24E+02	3.03E+01	2.75E+02	1.73E+02	1.85E+00
f_{28}	Mean	9.93E+02	9.66E+02	8.93E+02	1.18E+03	**7.78E+02**	2.13E+03	1.36E+03	8.88E+02
	Std	6.20E+01	6.11E+01	4.39E+01	2.75E+02	**1.93E+01**	6.60E+02	1.59E+02	3.32E+01
f_{29}	Mean	1.61E+03	1.09E+03	1.64E+03	1.28E+03	1.82E+03	5.00E+06	9.02E+03	**9.37E+02**
	Std	3.60E+02	1.11E+02	4.76E+02	3.65E+02	2.74E+02	4.43E+06	1.16E+04	**4.67E+00**
f_{30}	Mean	2.10E+03	2.97E+03	1.43E+03	4.30E+03	1.59E+03	5.81E+04	1.72E+03	**1.36E+03**
	Std	4.85E+02	6.66E+02	5.44E+02	2.43E+03	4.72E+02	2.90E+04	5.46E+02	**2.48E+02**

Table 7
Wilcoxon signed-rank test results.

	$p-value$	R^+	R^-	$n/w/t/l$
MEGWO VS ACS	0.0005	436	29	30/28/2/0
MEGWO VS MEABC	0.0012	431	34	30/26/3/1
MEGWO VS SinDE	0.0015	393	72	30/26/4/0
MEGWO VS BHS	0.0001	451	14	30/28/2/0
MEGWO VS DE/BBO	0.003	376	89	30/24/6/0
MEGWO VS WOA	0.0001	456	9	30/29/1/0
MEGWO VS CLPSO-LOT	0.001	374	91	30/21/7/2

the original global best strategy in GHS, the enhanced global-best lead strategy embeds an elite guidance operator which integrates social cognition into the experience of the alpha wolf. This modification supports the detailed local search around the current optimal solution and provide fast convergence to optimum or near optimum solutions. Secondly, the one-dimensional update operation adopted in the adaptable cooperative hunting strategy can avoid the destruction of excellent solution to some extent and improve the population diversity. The alternative operation of one-dimensional and total-dimensional update strategy can not only

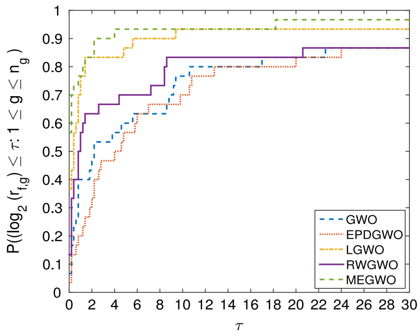


Fig. 4. Performance profile of 5 algorithms on CEC2014 benchmark functions.

retain the characteristics of GWO, but also makes use of the one-dimensional update operation in dealing with multimodal and separable functions. Lastly, the dispersal foraging provides an effective global search to avoid falling into local stagnation. Meanwhile, the parameter self-regulation strategy is essential to balance the global search ability and local search ability for the optimization process. Given all the reasons above, the mutual cooperation among several update strategies can effectively improve the performance of GWO for various complex function optimization problems.

4.3. Computation complexity and time complexity analysis

4.3.1. Computation complexity analysis

Any optimization algorithms need to have low computational complexity, so that it can solve real-life optimization problems with less computational volume. In order to analyse the computational complexity of GWO and MEGWO, the step wise computational complexity in terms of worst case of computation time is calculated as follows:

In the population initialization stage, both GWO and MEGWO have computational complexity $O(N)$.

As for the main iterative loop, in GWO, the position vector of each search agent is updated so the computational complexity is $O(N \cdot D)$ for this step. After updating the position of the population, the fitness values of all search agents are evaluated in $O(N)$ time. Because both the update process and evaluation operation are terminated within the maximum number of iterations saying T, the computational complexity is multiplied by T. After summarizing all the complexities as discussed above, the total computation complexity in GWO is obtained as $O(T \cdot N \cdot D)$.

In the main iterative progress of MEGWO, the enhanced global-best lead operation randomly select one element from the solution vector for each search agent in $O(1)$ time and update all search agents in $O(N \cdot D)$ time. In adaptable cooperative hunting step, the one-dimensional update operation only changes one element for each solution vector so the computational complexity is $O(N)$ while the total-dimensional update operation has computational complexity $O(N \cdot D)$. Because the above two update operations conduct in an alternate-running mode, the computational complexity is still $O(N \cdot D)$ in terms of worst case. In the last step of the algorithm, part of search agents will be dispersed to a promising area of the search space with $O(N)$ computational complexity. As a result, the total computation complexity in MEGWO is $O(T \cdot N \cdot D)$.

Hence, from the perspective of the worst complexity, both algorithms have the same value $O(T \cdot N \cdot D)$.

4.3.2. Time complexity analysis

According to the guidelines of IEEE CEC2014 [50], we supplemented the time complexity analysis of GWO and MEGWO. The parameters T_0, T_1, and T_2 are defined the same as in IEEE CEC2014. T_0 indicates the computing time for running a specified test program given in CEC2014. T_1 evaluates the computing time for 2×10^5 evaluations of Function 18 with a certain dimension D only. T_2 represents the complete computing time for one algorithm with 2×10^5 evaluations of the same D-dimensional Function 18. $\widehat{T}_2$ is the average value of five T_2 values.

Table 8
Time complexity for GWO and MEGWO (/s).

		$D = 10$	$D = 30$	$D = 50$
T_0		0.09	0.09	0.09
T_1		0.13	0.26	0.57
$\widehat{T}_2$	GWO	3.59	4.46	5.69
	MEGWO	1.54	2.62	3.74
$(\widehat{T}_2 - T_1)/T_0$	GWO	39.13	47.66	58.03
	MEGWO	15.97	26.73	35.91

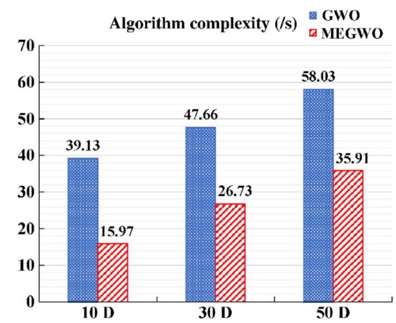


Fig. 5. Time complexity of GWO and MEGWO for dimensions 10, 30 and 50.

The time complexities of GWO and MEGWO for 10, 30, and 50-dimensional functions are shown in Table 8. The computing time $\widehat{T}_2$ of GWO is almost 2.3312 times of the one of MEGWO for 10-dimensional case. $\widehat{T}_2$ of GWO is approximately 70.23% more than the one of MEGWO for 30-dimensional case and 52.14% more than $\widehat{T}_2$ of MEGWO for 50-dimensional case. To better analyse the results, Fig. 5 is plotted. The figure shows that the computing complexity of GWO is obviously higher than that of MEGWO. The reason is that enhanced global-best lead strategy and adaptable cooperative hunting strategy in proposed MEGWO run alternately. This mechanism does not increase the time complexity of MEGWO in essence. On the other hand, MEGWO adopts parallel operations to evaluate the fitness value of the population while GWO utilizes serial operations. Therefore, MEGWO is superior to GWO in terms of time complexity.

4.4. Application to feature selection

In order to validate the performance of MEGWO on real-world problems, the proposed MEGWO was applied on feature selection (FS) which is considered as a multi-objectives optimization problem. The target of FS is to find the optimal feature subsets. FS can eliminate irrelevant or redundant features, so as to reduce the number of features, improve model accuracy and reduce running time [61]. The GWO was originally designed to solve problems with continuous variables. Due to the property of FS that solutions are restricted to the binary {0, 1}, the MEGWO should be developed into a binary version. Herein, Value "1" represents the corresponding feature is selected; otherwise the Value is set to"0". In this paper, firstly we allow each search agent update their positions in the continuous search spaces [0, 1] to a d-dimensional candidate solution v_i^t, and then each element of the solution v_i^t is converted to the binary {0, 1} based on a hard decision operator as below.

$$\begin{cases} x_{i,d}^t = 1, & v_{i,d}^t > 0.5 \\ x_{i,d}^t = 0, & \text{otherwise} \end{cases} \tag{18}$$

Table 9
List of datasets used in the experiment.

No	Dataset	No. of Samples	No. of Features	No. of Classes
1	Australian	690	14	2
2	Breast	277	9	2
3	Hearts	270	12	2
4	Ionosphere	351	34	2
5	Kr_vs_kp	3196	36	2
6	Sonar	208	60	2
7	Segmentation	210	19	7
8	Vowel	528	10	2
9	Wine	178	13	3
10	Waveform	5000	21	3
11	Wdbc	569	30	2
12	Zoo	101	16	7

The tradeoff of FS is between minimizing the number of selected features and maximizing classification accuracy. As a result, the objective optimization function can be expressed as [64]:

$$fitness = w_1 \cdot r_R + w_2 \cdot (|R|/|N|) \tag{19}$$

where r_R represents the classification error rate of a given classifier, $|R|$ indicates the size of the selected features subset and $|N|$ represents the number of the features in the dataset. w_1 and w_2 are two weighting coefficients corresponding to classification quality and subset size respectively. $w_1 \in [0, 1]$ and $w_2 = (1-w_1)$ are adopted in [64]. In our experiments, w_1 was set to 0.99 and w_2 was set to 0.01 which are commonly used in the literature [18,20].

Twelve datasets listed in Table 9 from the UCI machine learning repository [65] are performed in the experiments. These datasets have various numbers of samples, features and classifications. Each dataset was divided in a cross-validation manner [66]. In K-fold cross-validation, $K-1$ folds are used for training, validation and the remaining folds are used for testing. The size of dataset for training, validation, and testing was equal to each other. A wrapper approach-based on the K-Nearest Neighbour (KNN) classifier (where $K = 5$) was used for FS [67]. SRPSO [60], SinDE [57], and WOASA [61] were selected as the comparison algorithms to investigate the performance of MEGWO. For all experiments in this section, the population size for SRPSO, SinDE, and WOASA were all chosen as 10. The population size for MEGWO was set to 5. The maximum number of iterations for four algorithms was set to 100 and all algorithms ran 5 times.

Table 10 outlines the two main objectives, classification accuracy and average selection size for FS. Observed from Table 10, it is evident that MEGWO is much better than SRPSO on all datasets. With respect to SinDE, MEGWO outperforms it on almost all datasets except for Waveform dataset. For waveform dataset, SinDE provides 78.98% classification accuracy by using 15 features while MEGWO renders 78.82% accuracy with 16 features. Compared with WOASA, MEGWO obtains better results on 10 datasets in terms of classification accuracy except Sonar and Waveform datasets. Meanwhile, MEGWO provides better or similar results on 11 datasets in terms of the number of selected features except for Sonar dataset.

Table 11 shows the mean fitness values and the average computational time (/s) obtained from the four optimization algorithms. As can be seen in Table 11, MEGWO outperforms SRPSO, SinDE, and WOASA in terms of the mean fitness values on almost all the datasets except for Sonar and Waveform datasets. For Sonar dataset and Waveform dataset, WOASA obtained the best mean fitness values in comparison to other algorithms. With respect to the average computational time, MEGWO has the minimum computational time compared with other approaches except for Waveform dataset. For Waveform dataset, SRPSO has the minimum computational time in 64.83 s.

According to Tables 10 and 11, the performance ranking of the four algorithms is shown as: SRPSO < SinDE < WOASA < MEGWO. For Sonar and Waveform datasets which have large numbers of features or samples, WOASA can provide better results than MEGWO in terms of mean fitness values, classification accuracy and number of selected features. The reason is that the combination of WOA and Simulated Annealing (SA) allows the algorithm to explore extensively in the feature space and intensify the local search. Though the excessive execution of the embedded SA algorithm makes WOASA perform well for high dimensional optimization problems, this method consumes too many computational resources and increases time cost. In contrast, the performance results of MEGWO prove its capability to achieve a better balance between the accuracy and time complexity. The main reason is that, the one-dimensional update operation in MEGWO only changes one element of candidate solution vector in each iteration and employs the greedy selection strategy to ensure the new better solution obtained to be reserved for the next iteration. The operation can prevent the destruction of excellent solution to some extent and improve the population diversity. This characteristic allows MEGWO to work efficiently on multimodal and separable functions especially for feature selection problems.

5. Conclusion and future direction

In this study, an efficient GWO-based optimizer with several different update strategies, named as MEGWO, was proposed for various complex function optimization. In MEGWO, search agents update their positions based on the cooperation among three search strategies, which improves the global and local search ability of the proposed algorithm. MEGWO not only retains the advantage of fast convergence speed and strong local search ability of GWO, but also makes full use of the characteristics of each search strategy to balance the global and local search ability. The performance of the proposed algorithm was examined on 18 test functions and IEEE CEC2014 test set. Furthermore, feature selection was employed to investigate the effectiveness of MEGWO on real world applications. The obtained results show that the modification on original GWO accelerates convergence speed, enhances searching efficiency and improves the computational precision as well. It is also proved that the proposed MEGWO is an efficient and reliable algorithm for real world optimization problems.

Although the method presented in this paper effectively enhances the performance of GWO for optimization of various complex functions, it introduces more additional parameters. The users who wish to adopt MEGWO need to train the parameters to obtain the optimal setting in advance. Therefore, parameter-free GWO or parameter self-adaptive GWO to various real-world problems will be a vital research direction in our future work. Also, to the best of our knowledge, there is little work which consider about the weighting factors of alpha, beta and delta in Eq. (9). These factors reflect the contribution of each leader in the final solution and hence worth investigating in the future.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China under Grant No. 61301181, No. 61572534, and No. 61873290, in part by the Project-sponsored by SRF, China for ROCS, SEM, under Grant [2014]1685, and in part by the Special Project for Promoting Economic Development in Guangdong Province, China under Grant GDME-2018D004.

Conflict of interest

The authors declare that they have no conflict of interest.

Ethical approval

This article does not contain any studies with human participants performed by any of the authors.

Table 10
Classification accuracy and average number of selected features for SRPSO, SinDE, WOASA, and MEGWO.

Dataset	Accuracy				Average number of selected features			
	SRPSO	SinDE	WOASA	MEGWO	SRPSO	SinDE	WOASA	MEGWO
Australian	80.93%	85.39%	86.67%	**87.54%**	4.6	4.8	4.2	**3.2**
Breast	73.81%	78.99%	79.71%	**79.86%**	4.4	4.2	4.4	**4.0**
Hearts	76.00%	81.78%	85.19%	**85.33%**	4.8	4.6	4.2	**4.0**
Ionosphere	90.45%	94.55%	95.45%	**95.91%**	16.4	10.8	**10.6**	**10.6**
Kr_vs_kp	78.20%	80.23%	82.17%	**82.53%**	18.6	16.4	**15.2**	**15.2**
Sonar	86.15%	93.08%	**95.67%**	94.81%	28.2	26.4	**23.4**	25.6
Segmentation	88.76%	91.81%	91.43%	**94.67%**	9.8	7.0	8.0	**5.2**
Vowel	92.27%	96.52%	97.73%	**98.48%**	7.8	7.6	7.4	**7.0**
Wine	92.36%	95.53%	97.51%	**98.43%**	6.6	4.6	4.2	**4.0**
Waveform	75.54%	78.98%	**79.52%**	78.82%	18.2	**15**	16.0	16.0
Wdbc	92.21%	92.77%	94.04%	**94.53%**	5.2	5.4	5.2	**5.0**
Zoo	96.86%	98.04%	98.04%	**99.22%**	9.4	6.5	5.6	**5.4**

Table 11
Mean fitness value and average computational time (/s) for SRPSO, SinDE, WOASA, and MEGWO.

Dataset	Mean Fitness				Average Computational time (/s)			
	SRPSO	SinDE	WOASA	MEGWO	SRPSO	SinDE	WOASA	MEGWO
Australian	0.192	0.148	0.135	**0.126**	3.74	3.48	18.09	**3.46**
Breast	0.264	0.211	0.206	**0.204**	2.83	2.93	11.72	**2.62**
Hearts	0.242	0.184	0.157	**0.149**	2.84	2.84	16.35	**2.77**
Ionosphere	0.099	0.057	0.048	**0.044**	3.16	3.12	14.36	**2.89**
Kr_vs_kp	0.221	0.199	0.181	**0.177**	53.12	45.22	230.21	**38.67**
Sonar	0.142	0.072	**0.047**	0.056	4.38	3.85	11.84	**2.73**
Segmentation	0.117	0.0849	0.089	**0.0557**	3.42	3.49	26.76	**3.37**
Vowel	0.083	0.041	0.032	**0.022**	4.87	5.46	25.28	**3.51**
Wine	0.081	0.048	0.028	**0.019**	3.77	4.15	29.74	**3.71**
Waveform	0.248	0.215	**0.212**	0.217	**64.83**	79.94	368.76	84.82
Wdbc	0.082	0.073	0.061	**0.056**	5.58	4.56	25.16	**4.52**
Zoo	0.037	0.023	0.023	**0.011**	2.83	2.95	16.17	**2.71**

References

[1] G. Wang, X. Cai, Z. Cui, G. Min, J. Chen, High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm, IEEE Trans. Emerg. Top. Comput. PP (99) (2017) http://dx.doi.org/10.1109/TETC.2017.2703784, 1–1.

[2] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680, http://dx.doi.org/10.1126/science.220.4598.671.

[3] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[4] P. Heermann, N. Khazenie, Classification of multispectral remote sensing data using a back-propagation neural network, IEEE Trans. Geosci. Remote Sens. 30 (1) (1992) 81–88, http://dx.doi.org/10.1109/36.124218.

[5] R. Storn, K. Price, Differential Evolution - A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359, http://dx.doi.org/10.1023/A:1008202821328.

[6] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proc. of 1995 IEEE Int. Conf. Neural Networks, Vol. 4, Perth, Australia), Nov. 27–Dec. (8) 2011, pp. 1942–1948, http://dx.doi.org/10.1007/978-0-387-30164-8_630.

[7] X. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84, http://dx.doi.org/10.1504/IJBIC.2010.032124.

[8] W. Zong, J. Kim, G. Loganathan, A new heuristic optimization algorithm: Harmony search, Simulation 2 (2) (2001) 60–68, http://dx.doi.org/10.1177/003754970107600201.

[9] X. Yang, A. Gandomi, Bat algorithm: A novel approach for global engineering optimization, Eng. Comput. 29 (5) (2012) 464–483, http://dx.doi.org/10.1108/02644401211235834.

[10] A. Gandomi, X. Yang, A. Alavi, Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35, http://dx.doi.org/10.1007/s00366-011-0241-y.

[11] S. Mishra, A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation, IEEE Trans. Evol. Comput. 9 (1) (2005) 61–73, http://dx.doi.org/10.1109/TEVC.2004.840144.

[12] M. Eusuff, K. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, J. Water Resour. Plan. Manag. 129 (3) (2003) 210–225, http://dx.doi.org/10.1061/(ASCE)0733-9496(2003)129:3(210).

[13] D. Karaboga, B. Akay, A comparative study of Artificial Bee Colony algorithm, Appl. Math. Comput. 214 (1) (2009) 108–132, http://dx.doi.org/10.1016/j.amc.2009.03.090.

[14] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713, http://dx.doi.org/10.1109/TEVC.2008.919004.

[15] S. Mirjalili, S. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (3) (2014) 46–61, http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

[16] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

[17] H. Faris, M. Mafarja, A. Heidari, I. Aljarah, A. Al-Zoubi, An efficient binary salp swarm algorithm with crossover scheme for feature selection problems, Knowl.-Based Syst. 154 (2018) 43–67, http://dx.doi.org/10.1016/j.knosys.2018.05.009.

[18] M. Mafarja, I. Aljarah, A. Heidari, A. Hammouri, Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems, Knowl.-Based Syst. 145 (2018) 25–45, http://dx.doi.org/10.1016/j.knosys.2017.12.037.

[19] H. Faris, A. Ala'M, A. Heidari, I. Aljarah, M. Mafarja, An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks, Inf. Fusion 48 (2019) 67–83, http://dx.doi.org/10.1016/j.inffus.2018.08.002.

[20] I. Aljarah, M. Mafarja, A. Heidari, H. Faris, Y. Zhang, Asynchronous accelerating multi-leader salp chains for feature selection, Appl. Soft Comput. 71 (2018) 964–979, http://dx.doi.org/10.1016/j.asoc.2018.07.040.

[21] M. Mafarja, I. Aljarah, A. Heidari, H. Faris, Binary dragonfly optimization for feature selection using time-varying transfer functions, Knowl.-Based Syst. (2018) http://dx.doi.org/10.1016/j.knosys.2018.08.003.

[22] H. Faris, I. Aljarah, M. Al-Betar, S. Mirjalili, Grey wolf optimizer: A review of recent variants and applications, Neural Comput. Appl. 30 (2) (2018) 413–435, http://dx.doi.org/10.1007/s00521-017-3272-5.

[23] E. Emary, H. Zawbaa, A. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381, http://dx.doi.org/10.1016/jneucom.2015.06.083.

[24] M. Sulaiman, Z. Mustaffa, M. Mohamed, O. Aliman, Using the gray wolf optimizer for solving optimal reactive power dispatch problem, Appl. Soft Comput. 32 (2015) 286–292, http://dx.doi.org/10.1016/j.asoc.2015.03.041.

[25] N. Al-Aboody, H. Al-Raweshidy, Grey wolf optimization-based energy-efficient routing protocol for heterogeneous wireless sensor networks, in: Computational and Business Intelligence (ISCBI), 2016 4th International Symposium on, IEEE, 2016, pp. 101–107, http://dx.doi.org/10.1109/ISCBI.2016.7743266.

[26] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, Grey wolf optimizer for parameter estimation in surface waves, Soil Dyn. Earthq. Eng. 75 (2015) 147–157, http://dx.doi.org/10.1016/j.soildyn.2015.04.004.

[27] A. Sahoo, S. Chandra, Multi-objective grey wolf optimizer for improved cervix lesion classification, Appl. Soft Comput. 52 (2017) 64–80, http://dx.doi.org/10.1016/j.asoc.2016.12.022.
[28] A. Khairuzzaman, S. Chaudhury, Multilevel thresholding using grey wolf optimizer for image segmentation, Expert Syst. Appl. 86 (2017) 64–76, http://dx.doi.org/10.1016/j.eswa.2017.04.029.
[29] M. Malik, E. Mohideen, L. Ali, Weighted distance grey wolf optimizer for global optimization problems, in: 2015 IEEE International Conference on Computational Intelligence & Computing Research (ICCIC), 2015, pp. 1–6, http://dx.doi.org/10.1109/ICCIC.2015.7435714.
[30] L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, A fuzzy hierarchical operator in the grey wolf optimizer algorithm, Appl. Soft Comput. 57 (2017) 315–328, http://dx.doi.org/10.1016/j.asoc.2017.03.048.
[31] M. Al-Betar, M. Awadallah, H. Faris, I. Aljarah, Natural selection methods for grey wolf optimizer, Expert Syst. Appl. 113 (2018) 481–498, http://dx.doi.org/10.1016/j.eswa.2018.07.022.
[32] S. Saremi, S. Mirjalili, S. Mirjalili, Evolutionary population dynamics and grey wolf optimizer, Neural Comput. Appl. 26 (5) (2015) 1257–1263, http://dx.doi.org/10.1007/s00521-014-1806-7.
[33] B. Mahdad, K. Srairi, Blackout risk prevention in a smart grid based flexible optimal strategy using grey wolf-pattern search algorithms, Energy Convers. Manage. 98 (2015) 411–429, http://dx.doi.org/10.1016/j.enconman.2015.04.005.
[34] J. Zhou, W. Zhu, Y. Zheng, C. Li, Precise equivalent model of small hydro generator cluster and its parameter identification using improved Grey Wolf optimizer, IET Gener. Transm. Distrib. 10 (9) (2016) 2108–2117, http://dx.doi.org/10.1049/iet-gtd.2015.1141.
[35] A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with lévy flight for optimization tasks, Appl. Soft Comput. 60 (2017) 115–134, http://dx.doi.org/10.1016/j.asoc.2017.06.044.
[36] B. Yang, X. Zhang, T. Yu, H. Shu, Z. Fang, Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine, Energy Convers. Manag. 133 (2017) 427–443, http://dx.doi.org/10.1016/j.enconman.2016.10.062.
[37] S. Gupta, K. Deep, A novel random walk grey wolf optimizer, Swarm Evolut. Comput. (2018) http://dx.doi.org/10.1016/j.swevo.2018.01.001.
[38] V. Kamboj, A novel hybrid PSO-GWO approach for unit commitment problem, Neural Comput. Appl. 27 (6) (2016) 1643–1655, http://dx.doi.org/10.1007/s00521-015-1962-4.
[39] A. Zhu, C. Xu, Z. Li, J. Wu, Z. Liu, Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC, J. Syst. Eng. Electron. 26 (2) (2015) 317–328, http://dx.doi.org/10.1109/JSEE.2015.00037.
[40] S. Mirjalili, How effective is the grey wolf optimizer in training multi-layer perceptrons, Appl. Intell. 43 (1) (2015) 150–161, http://dx.doi.org/10.1007/s10489-014-0645-7.
[41] C. Muro, R. Escobedo, L. Spector, R. Coppinger, Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations, Behav. Process. 88 (3) (2011) 192–197, http://dx.doi.org/10.1016/j.beproc.2011.09.006.
[42] P. Suganthan, N. Hansen, J. Liang, K. Deb, Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization, KanGAL report, 2005005, Nanyang Technological University Singapore, 2005, p. 2005.
[43] M. Omran, M. Mahdavi, Global-best harmony search, Appl. Math. Comput. 198 (2) (2008) 643–656, http://dx.doi.org/10.1016/j.amc.2007.09.004.
[44] W. Xiang, M. An, Y. Li, R. He, J. Zhang, An improved global-best harmony search algorithm for faster optimization, Expert Syst. Appl. 41 (13) (2014) 5788–5803, http://dx.doi.org/10.1016/j.eswa.2014.03.016.
[45] W. Gao, S. Liu, L. Huang, A global best artificial bee colony algorithm for global optimization, J. Comput. Appl. Math. 236 (11) (2012) 2741–2753, http://dx.doi.org/10.1016/j.cam.2012.01.013.
[46] M. Kiran, H. Hakli, M. Gunduz, H. Uguz, Artificial bee colony algorithm with variable search strategy for continuous optimization, Inform. Sci. 300 (2015) 140–157, http://dx.doi.org/10.1016/j.ins.2014.12.043.
[47] Y. Shi, C. Pun, H. Hu, H. Gao, An improved artificial bee colony and its application, Knowl.-Based Syst. 107 (2016) 14–31, http://dx.doi.org/10.1016/j.knosys.2016.05.052.
[48] W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Trans. Cybern. 43 (3) (2013) 1011–1024, http://dx.doi.org/10.1109/TSMCB.2012.2222373.
[49] X. Li, M. Yin, Modified cuckoo search algorithm with self adaptive parameter method, Inform. Sci. 298 (2015) 80–97, http://dx.doi.org/10.1016/j.ins.2014.11.042.
[50] J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.
[51] M. Cheng, D. Prayogo, Symbiotic organisms search: A new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112, http://dx.doi.org/10.1016/j.compstruc.2014.03.007.
[52] X. Zhang, Q. Kang, J. Cheng, X. Wang, A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer, Appl. Soft Comput. 67 (2018) 197–214, http://dx.doi.org/10.1016/j.asoc.2018.02.049.
[53] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evolut. Comput. 1 (1) (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002.
[54] W. Gong, Z. Cai, C. Ling, DE/BBO: A hybrid differential evolution with biogeography-based optimization for global numerical optimization, Soft Comput. 15 (4) (2010) 645–665, http://dx.doi.org/10.1007/s00500-010-0591-1.
[55] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, J. Pan, Multi-strategy ensemble artificial bee colony algorithm, Inform. Sci. 279 (2014) 587–603, http://dx.doi.org/10.1016/j.ins.2014.04.013.
[56] M. Naik, M. Nath, A. Wunnava, A new adaptive cuckoo search algorithm in: 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), Vol. 19, 2015, pp. 1–5. http://dx.doi.org/10.1109/ReTIS.2015.7232842.
[57] A. Draa, S. Bouzoubia, I. Boukhalfa, A sinusoidal differential evolution algorithm for numerical optimisation, Appl. Soft Comput. 27 (2015) 99–126, http://dx.doi.org/10.1016/j.asoc.2014.11.003.
[58] Y. Zheng, M. Zhang, B. Zhang, Biogeographic harmony search for emergency air transportation, Soft Comput. 20 (3) (2016) 967–977, http://dx.doi.org/10.1007/s00500-014-1556-6.
[59] K. Zhang, Q. Huang, Y. Zhang, Enhancing comprehensive learning particle swarm optimization with local optima topology, Inform. Sci. 471 (2019) 1–18, http://dx.doi.org/10.1016/j.ins.2018.08.049.
[60] M. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inform. Sci. 294 (2015) 182–202, http://dx.doi.org/10.1016/j.ins.2014.09.053.
[61] M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312, http://dx.doi.org/10.1016/j.neucom.2017.04.053.
[62] E. Dolan, J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2) (2002) 201–213, http://dx.doi.org/10.1007/s101070100263.
[63] R. Blair, J. Higgins, A comparison of the power of wilcoxon's rank-sum statistic to that of student's t statistic under various nonnormal distributions, J. Educ. Stat. 5 (4) (1980) 309–335, http://dx.doi.org/10.3102/10769986005004309.
[64] E. Emary, H. Zawbaa, A. Hassanien, Binary ant lion approaches for feature selection, Neurocomputing 213 (2016) 54–65, http://dx.doi.org/10.1016/j.neucom.2016.03.101.
[65] C. Blake, UCI repository of machine learning databases, 1998, University of California, Department of Information and Computer Science, http://archive.ics.uci.edu/ml/index.php.
[66] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010) 40–79, http://dx.doi.org/10.1214/09-SS054.
[67] Z. Liu, Q. Pan, J. Dezert, A new belief-based k-nearest neighbor classification method, Pattern Recognit. 46 (3) (2013) 834–844, http://dx.doi.org/10.1016/j.patcog.2012.10.001.