



Research article

Multimodal optimization using whale optimization algorithm enhanced with local search and niching technique

Hui Li*, Peng Zou, Zhiguo Huang, Chenbo Zeng, and Xiao Liu

Department of Computer Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

* **Correspondence:** ray@mail.buct.edu.cn.

Abstract: For some real-world problems, it is desirable to find multiple global optima as many as possible. The multimodal optimization approach which finds multiple optima in a single run shows significant difference with the single modal optimization approach. The whale optimization algorithm (WOA) is a newly emerging reputable optimization algorithm. Its global search ability has been verified in many benchmark functions and real-world applications. In this paper, we propose a multimodal version of whale optimization algorithm (MMWOA). MMWOA enhances the multimodal search ability of WOA by using the niching technique and improves the local search efficiency of WOA by combining the Gaussian sampling technique. The algorithm has been tested on multimodal optimization benchmark functions recommended by CEC'2013 and on a multimodal optimization problem with non-linear constraints. Experimental results indicate that MMWOA has competitive performance compared with other state-of-the-art multimodal optimization algorithms.

Keywords: whale optimization algorithm; local search; clustering; niching algorithms

1. Introduction

Traditional mathematical methods such as gradient method are deterministic and work effectively for unimodal optimization. These algorithms show a dramatic increase in computational effort for the multimodal problems. Nature inspired meta-heuristic algorithms can handle these multimodal problems more efficiently. The advantage is even greater when little information regarding the model can be obtained and brute force cannot be used due to the vast search space.

Various and diverse animals and plants develop their strategies over millions of years or more to keep survival. Nature inspired algorithms derive their model from these successful species. Swarm algorithms, as a typical nature inspired meta-heuristic, such as ant colony optimization, particle swarm optimization, fish swarm optimization and many more, are based on the collaborative social

behaviour of animals. Swarm algorithms exhibit many characteristics: Firstly, since most metaheuristics are inspired by very simple concepts, they can be implemented without much difficulty. Secondly, most metaheuristics assume problems as black boxes, so they have the ability to be applied to various problems without much changes in the structure of the algorithm. Lastly, metaheuristics have natural ability to avoid local optima compared with traditional optimization algorithms such as gradient-based algorithms.

Kennedy and Eberhart [1] proposed the particle swarm optimization in 1995 which is related to the flocking of birds and the behaviour of fish schools. Yang and Deb [2] presented the cuckoo search algorithm based on the parasitic nature of the cuckoo. The cuckoo lays its eggs in other birds' nests thus releasing itself from the burden of parenting. The firefly algorithm arose from the work of Xin-She Yang in 2008 [3], which adapts the behaviour of firefly swarms to develop an optimization algorithm. A common belief among researchers is that almost all swarm intelligence algorithms have both exploratory and exploitative behaviour. Exploration is a behaviour to avoid premature convergence to local optima, whereas exploitation is a behaviour to conduct a local search so as to get closer to a possible optimum. Different algorithms have different strategies to maintain the balance between exploration and exploitation.

In 2016, Mirjalili and Lewis [4] proposed the whale optimization algorithm (WOA) which mimics the hunting strategy of humpback whales. WOA simulates the bubble-net foraging behaviour of humpback whales including searching for prey, encircling the prey and approaching the prey. It has been reported in [4] that WOA has competitive search ability. Moreover, there is vast literature providing many successful applications in the past two years.

WOA has been successfully used to improve feature selection accuracy by determining the optimal boundary for the classification and clustering processes [5–7]. Thresholding of images separates the interesting objects from their background. In image processing, thresholding can be optimized by WOA to improve the segmentation accuracy such as liver segmentation in MRI images [8] and segmentation of the retinal fundus images [9]. Pattern recognition can also be implemented with the aid of WOA. Methods based on WOA have been proposed to effectively identify the compound faults of bearing and gearbox in [10] and [11] respectively. Parameter estimation is another typical application of WOA. Oliva [12] proposed WOA based method to estimate parameters of solar cells. Elazab et al. [13] estimate parameters of various PV models using WOA. Ren et al. [14] proposed a hybrid improvement algorithm based on WOA to estimate parameters in Lorenz system and Lu system. WOA has been successfully applied to the control systems. Medani et al. [15] used WOA to find the best control vector of control variables for minimizing active power losses. Simhadri et al. [16] employed WOA as a meta-heuristic for automatic generation control problem. Yousri et al. [17] proposed a method to control chaotic behaviour of the permanent magnet synchronous motor.

WOA can help obtain the approximate optimum for some NP-hard problems. Abdel et al. [18] presented an improved WOA for solving both single and multidimensional 0–1 knapsack problems with different scales. Ghahremaninahr et al. [19] provided a WOA based solution for a multi-echelon multi-product facility location/allocation problem. Wang et al. [20] solved a wireless sensor network coverage optimization problem using WOA based methods. More and more interesting applications of WOA appeared recently such as training of neural network [21], cryptanalysis [22], and circuit design [23].

Various hybrid algorithms that integrate the WOA with other meta-heuristic algorithms have been

proposed in the past two years. These algorithms encompass integrating WOA with particle swarm optimization (PSO) [24, 25], integrating WOA with differential evolution (DE) algorithm [13, 26], integrating WOA with tabu search [27], integrating WOA with simulated annealing (SA) algorithm [28, 29], integrating WOA with gray wolf optimization (GWO) [23], and integrating WOA with krill-herd (KH) algorithm [30]. Additionally, there are some variants of WOA in recent literature. Emary proposed a variant based on levy flight [31]. Khalil et al. propose a distributed implementation of WOA using hadoop mapreduce named MR-WOA [32].

According to the comparative studies presented in [4], WOA algorithm appears to be very effective in particular for its fast convergence to the global optimum as well as its significant active loss reduction. Although WOA has been successfully employed in various optimization applications, there is no WOA based solution for multimodal optimization problems has been reported so far.

The goal of locating multiple optimal solutions in a single run, referred to as multimodal optimization, contrasts sharply with the goal of a classic optimization method which usually starts from an initial single point and iteratively improving it before arriving at one final solution [33]. Traditional niching methods, including fitness sharing and crowding methods, were developed in the early 70s and 80s [34]. In subsequent years, many niching methods have been proposed. Developing multimodal approaches based on meta-heuristic optimization algorithms is the latest trend on multimodal optimization [35].

Brits et al. [36] proposed a NichePSO method to locate and refine multiple solutions. Qu et al. [37] presented a multimodal DE algorithm with a neighbourhood mutation strategy. Nekouie and Yaghoobi [38] proposed multimodal optimization based on firefly algorithm. The multimodal optima are detected through separately evolving sub-populations. Banati and Chaudhary [39] developed a multimodal variant of bat algorithm (BA) with improved search capabilities. Galvez et al. [40] proposed an algorithm named multimodal flower pollination algorithm (MFPA) and evaluate it in a set of benchmark functions.

As no free lunch (NFL) theorem [41] claimed, there is no meta-heuristic best suited for solving all optimization problems. So, developing various multimodal metaheuristics algorithms becomes an active research area in recent years. As mentioned earlier, WOA has competitive search ability which has been justified in many benchmark functions and real-world applications. Hence it motivates us to extend WOA algorithm to deal with multimodal optimization problems and propose the multimodal version of WOA in this paper. In the multimodal whale optimization algorithm (MMWOA), we introduce the niching technique to enhance the multimodal search ability. The K-means clustering method and the clustering method with fixed cluster size bound are two main clustering method used for niching. Furthermore, we combine the local search with WOA to improve the exploitation efficiency of MMWOA.

To verify the efficiency and effectiveness of the proposed MMWOA, extensive experiments on widely used benchmark problems are conducted. The benchmark problems encompass 20 unconstrained global optimization problems from CEC'2013 and a nonlinear constrained optimization problem [42, 43]. We also make fair comparisons with some state-of-the-art multimodal optimization algorithms. Specifically, SDE and LIPS are two reputable multimodal algorithms which outperform many other algorithms. SDE [44] is an extension to the basic DE using speciation [45], which is more computationally efficient than crowding-DE [46], especially when a large population size has to be used to address the multimodal problem with a large number of optima. LIPS [47] is a

distance-based locally informed particle swarm optimizer, which uses several local bests to guide the search of each particle instead of using the global best particle. SDE and LIPS are widely used as baseline multimodal optimization algorithms [48]. Hence we measure the performance of MMWOA on benchmark problems from CEC'2013 and compare them with SDE and LIPS.

As a competitive multimodal optimization algorithm, MMWOA outperforms SDE and LIPS for low dimension benchmark functions such as F1–F5, and behaves well for some of very high dimension benchmark functions such as F16 and F18. However, it does not mean that MMWOA is better than SDE and LIPS. For one thing, MMWOA does not show advantages at all accuracy levels for F1–F5. For another, LIPS shows distinct advantages against MMWOA for benchmark functions with three variables. Therefore, more effort should be made to improve the performance of MMWOA, such as the performance for functions with three variables.

The organization of the paper is as follows. In section 2, we introduce the original WOA algorithm. Section 3 proposes the multimodal whale optimization algorithm and describes it in detail. We present the experimental setup and corresponding results in section 4, giving the list of benchmark functions used, explaining the evaluation criteria, and discussing the results. Section 5 concludes the paper finally.

2. Whale optimization algorithm

Proposed in 2016 by Mirjalili and Lewis [4], whale optimization algorithm (WOA) is a metaheuristic search algorithm which mimics the hunting strategy of humpback whales. Humpback whales prefer to hunt krill and small fish herds. A gam of whales search for prey in a shrinking circle. When they find prey, they blow bubbles in a spiral shape around the prey and swim up toward the surface [49]. The hunting behaviour is called bubble-net feeding in literature [50].

In the following introduction of WOA, we use the term “swarm” or “population” as the counterpart of the whale gam, use the term “search agent” as the counterpart of the individual whale, in the algorithm respectively. The optimal solution in the single-modal optimization problem is the location of the prey. The objective of the optimization problem is letting the best agent arrive or approach the target (optimal solution) as closely as possible, where the best agent is the search agent with the highest fitness.

The WOA can be mathematically modelled in three phases including encircling prey, bubble-net attacking, and random search for prey. Encircling prey and bubble-net attacking can be regarded as exploitation, while random search can be regarded as exploration.

2.1. Encircling prey

WOA initially searches the optimal solution by calculating and comparing the fitnesses of the search agents. Since the optimal solution is not previously known, other humpback whales will encircle the best agent to update their positions. The behaviour can be formulated as follows:

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \quad (2.1)$$

where \vec{X}^* is the position vector of the best solution obtained so far, \vec{X} is the position vector of search

agents, t represents the current iteration, $|$ is the absolute value and \cdot is an entry-wise multiplication, \vec{A} and \vec{C} are coefficient vectors and can be calculated as:

$$\begin{aligned}\vec{A} &= 2\vec{a} \cdot \vec{r} - \vec{a} \\ \vec{C} &= 2 \cdot \vec{r}\end{aligned}\quad (2.2)$$

where \vec{a} is linearly decreased from 2 to 0 over the course of search, and \vec{r} is a random vector which satisfies the uniform distribution in $[0, 1]$.

Eq. 2.1 allows any search agent to update its position in the neighbourhood of the current best solution and thus mimic encircling the prey.

2.2. Bubble-net attacking

Two procedures including shrinking encircling and spiral updating are designed to mathematically model the bubble-net attacking behaviour.

In shrinking encircling, the process is implemented by decreasing the value of \vec{a} in the Eq. 2.2 over the course of iterations. Restricting the value of \vec{A} to $[-1, 1]$, the new position of a search agent can be defined anywhere between the current position of the agent and the position of the best agent.

In the spiral updating process, a spiral equation is employed to mimic the helix-shaped movement of humpback whales as follows:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \quad (2.3)$$

where $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ defines distance of the i^{th} search agent to the target, b is a constant defining the shape of the logarithmic spiral, l is a random vector in uniform distribution $[-1, 1]$.

Since the humpback whales swim around the prey within a shrinking circle and along a spiral-shaped path simultaneously. WOA assumes that there is a probability of 0.5 to choose one of the above mentioned processes in each iteration. Corresponding mathematical formula is as follow:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)|, & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & \text{if } p \geq 0.5 \end{cases} \quad (2.4)$$

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

2.3. Random search for prey

Exploration is required for most nature inspired metaheuristics. In contrast to the exploitation phase, WOA updates the swarm as

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot |\vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X}(t)|, \quad (2.5)$$

where $\overrightarrow{X_{rand}}$ is a search agent randomly chosen from the current swarm, which does not have to be the best agent.

The value of coefficient A is used to determine whether to force an agent to perform a random search. When $|A| \geq 1$, the search agent is forced to explore.

At the end of this section, we summary the whole WOA algorithm in the flowchart as shown in Figure 1.

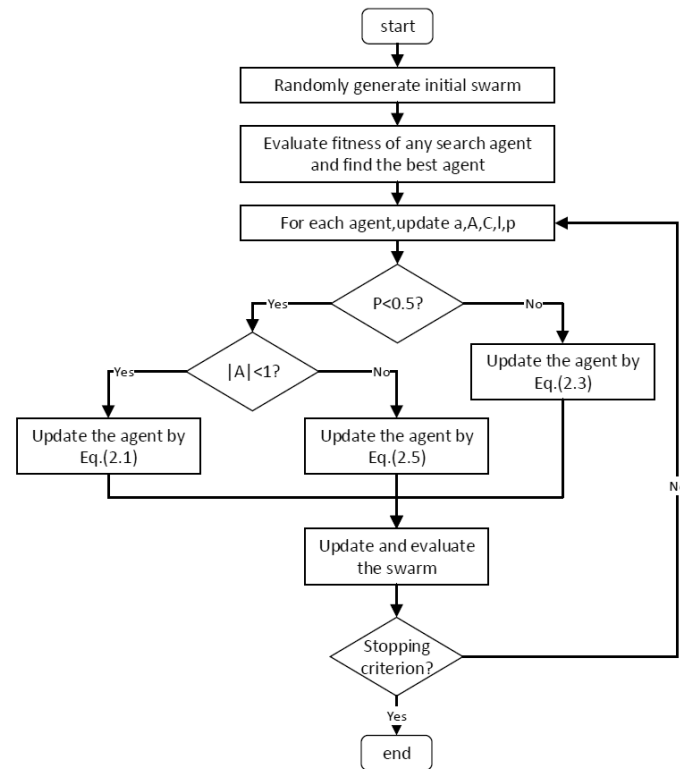


Figure 1. The flowchart of WOA.

3. Multimodal whale optimization algorithm

In this section, we propose a multimodal whale optimization algorithm (MMWOA) to address the multimodal optimization problem. The niching technique plays a crucial role in finding multiple optimal solutions. In this paper, we introduce clustering as a niching technique into the multimodal optimizer. Clustering allows the search to be performed at the niche level. Additionally, a local search scheme is embedded into MMWOA to promote exploitation for the MMWOA.

3.1. Niching strategies

Clustering is a process of dividing a data set into different groups so that the items in one group are similar to each other but not similar to the items in other groups. In this paper, we introduce the K -means clustering algorithm to address the multimodal problem. K -means clustering algorithm is one of the most easily implemented clustering algorithms. In K -means clustering algorithm, a given data set is assumed to contain n items. First, K items are randomly selected as the initial cluster centre. Next, the distance from each non-clustered centre item to each cluster centre is calculated. Then each item is assigned to a cluster which has the centre closest to the item. Once all items have been assigned, the centre of each cluster is recalculated based on all items in the cluster. The process is repeated until the stopping criteria are met.

The basic stopping criterion can be formulated as follows:

$$F = \sum_{j=1}^k \sum_{i \in cl(j)} \|x_i^j - c_j\|, \quad (3.1)$$

where $\|x_i^j - c_j\|$ is a distance measure between the item x_i^j in the j^{th} cluster $cl(j)$ and the centre c_j of the cluster. F is an indicator of the distance of the n items from their cluster centres. When the F reaches a given value or F no longer changes, the clustering process terminates.

The K-means clustering algorithm for niching is depicted in Algorithm 1.

Algorithm 1 K-means clustering algorithm for niching

Input: the initial swarm $X = \{x_1, x_2, \dots, x_n\}$ and the number of clusters (species) k

Output: k species

- 1: Randomly select K search agents from the swarm as the initial cluster centre $C = \{c_1, c_2, \dots, c_k\}$
 - 2: **for** j from 1 to k **do**
 - 3: **for** i from 1 to n **do**
 - 4: Calculate the distance from each search agent x_i to the cluster centre c_j ;
 - 5: **end for**
 - 6: **end for**
 - 7: Assign $x \in X$ to the cluster with the closest centre.
 - 8: **for** j from 1 to k **do**
 - 9: Update each cluster centre $c_j = \frac{1}{|c_j|} \sum_{x \in c_j} x$;
 - 10: **end for**
 - 11: Stop if the cluster centre no longer changes, otherwise go to 2.
-

We also introduce another clustering algorithm [51] which is widely used. In the clustering algorithm, the bound of each cluster size is set to a fixed value. Assuming that the size of the whole set is N , the fixed size bound of a cluster is m , then the number of clusters can be obtained as N/m . If m is not divide N , the remainder items are grouped into a new species, and thus the number of clusters is $N/m + 1$ in that case. The pseudo code of the fixed size bound clustering algorithm for niching is shown in Algorithm 2.

To facilitate the description of the algorithms, we name the MMWOA which absorbs the K-means clustering technique as K-MMWOA, and we name the MMWOA which using the clustering with a fixed cluster size bound as FS-MMWOA.

3.2. Generating new solutions using whale optimization algorithm

In MMWOA, each search agent in a species generates a new position for next step using Eq. 2.1, Eq. 2.3, and Eq. 2.5. Since the updating operation is performed within the species, the $\vec{X}^*(t)$ in Eq. 2.1 and Eq. 2.3 is not the best agent in the swarm of the t^{th} generation, but the best agent in a species. Likewise, the \vec{X}_{rand} in (2.5) is not a random agent in the swarm, but a search agent randomly selected from its species.

The pseudo code for generating new solutions is outlined in Algorithm 3.

Algorithm 2 Clustering algorithm with fixed cluster size bound for niching

Input: The swarm S and the species size m

Output: A set of species

```

1: while  $S$  is not empty do
2:   Randomly choose a search agent  $R$ 
3:   if The size of  $S$  is larger than  $m$  then
4:     Compute its distance to all search agents
5:     Select  $(m - 1)$  nearest search agents to  $R$  in  $S$  and form a species
6:   else
7:     Select all search agents to form a species
8:   end if
9:   Eliminate these search agents from  $S$ ;
10: end while

```

Algorithm 3 Generating new positions for agents using WOA

Input: The current swarm S

Output: The candidate positions of the swarm for next iteration S_c

```

1: for the  $i^{th}$  species in  $S$  do
2:   Obtain size of the species  $ss$ 
3:   Obtain the best agent in this species
4:   for  $X_j$  ( $j$  is from 1 to  $ss$ ) in the  $i^{th}$  species do
5:     Update  $a, A, C, l$  and  $p$ 
6:     if  $p < 0.5$  then
7:       if  $|A| < 1$  then
8:         Generate a new position using Eq. 2.1 and append it to  $S_c$ 
9:       else
10:        Select a random whale ( $X_{rand}$ ) in this species
11:        Generate a new position using Eq. 2.5 and append it to  $S_c$ 
12:       end if
13:     else
14:       Generate a new position using Eq. 2.3 and append it to  $S_c$ 
15:     end if
16:   end for
17: end for

```

3.3. Local Search

Most of the efficient hybrid optimization algorithms are produced by combination of the global optimization algorithm with the local search algorithm [52]. The local search is used to improve the solution quality gradually. Therefore, we introduce the gaussian sampling which is a local search technique to MMWOA. The gaussian distribution can be mathematically formulated as follows:

$$G(x^d, \mu^d, \delta) = \frac{1}{\delta \sqrt{2\pi}} e^{-\frac{(x^d - \mu^d)^2}{2\delta^2}}. \quad (3.2)$$

Since local search is performed near the current best agent in each species, the mean value μ of the Gaussian distribution is assigned to the position of the best agent. The standard deviation δ is set to a very small value. The parameter d represents the dimension of the problem.

An appropriate number of sampling points is needed to balance the computational cost and the opportunity to find better solutions. The number of the sampling points are set to four in this paper. Four is large enough to provide enough opportunity and small enough to avoid extra computational cost. If the sampling point has higher fitness than the current agent, it replaces the current agent with itself.

It is more efficient to conduct local search for agents near the optimal solutions. Those agents far away from the optimal solutions do not need to conduct local search. Moreover, to reduce the computation costs, only the best agent in each species conduct local search. For these reasons, we conduct local search for each best agent with a probability which is determined by its fitness. One can calculate the probability as follows [48]:

$$P_i = \frac{Fit_i + |Fit_{min}| + \eta}{Fit_{max} + |Fit_{min}| + \eta}, \quad (3.3)$$

where P_i is the probability to conduct local search for the i^{th} best agent, and Fit_i is its fitness. Fit_{max} and Fit_{min} are the maximum and minimum fitness among the best agents. It is possible for the fitness value to be zero or a negative number. So we introduce the item of Fit_{min} and η which is a very small positive value to avoid divide-by-zero. The pseudo code for local search is shown in Algorithm 4.

3.4. Summary of Multimodal whale optimization algorithm

MMWOA combines the WOA, the niching strategies, and the local search scheme. It can be divided into K-MMWOA and FS-MMWOA according to its clustering method. The pseudo code of K-MMWOA and FS-MMWOA are described in Algorithms 5 and 6, respectively.

There is an updating mechanism from step 4 to step 6 in both the K-MMWOA and the FS-MMWOA. In this updating mechanism, all agents update their positions as a whole. An alternative updating method is updating agents within the scope of each species.

3.5. Computational complexity

The complexity of the clustering procedure can be estimated based on the number of required calculations of Euclidean distances between two agents. Assume N is the swarm size and k is the number of clusters. k is a given number in K-MMWOA. For FS-MMWOA, k is equal to the ceiling of N/fs , where the ceiling of a number is the smallest integer greater than or equal to the number, and

Algorithm 4 Local Search

Input: Current best agent set S_b made up of the best agent of each species, its fitness set Fit_{sb} , the standard deviation δ , the number of sampling points N , and η

Output: Updated best agent set S_b

- 1: Let Fit_{min} be $\min(Fit_{sb})$, Fit_{max} be $\max(Fit_{sb})$
 - 2: Obtain $sizeOfSb$ which is the size of S_b
 - 3: **for** i from 1 to $sizeOfSb$ **do**
 - 4: **if** $rand() \leq pmf[i]$ **then**
 - 5: **for** j from 1 to N **do**
 - 6: Generate new position LS_j using Eq. 3.2
 - 7: Replace $S_b[i]$ with LS_j , if LS_j has higher fitness than $S_b[i]$
 - 8: **end for**
 - 9: **end if**
 - 10: **end for**
-

Algorithm 5 K-MMWOA

Input: the swarm size $sizeOfSwarm$, the maximum number of cluster(species) k

Output: the swarm with multiple best agents

- 1: Randomly initialize a swarm S with $sizeOfSwarm$ agents
 - 2: Using Algorithm 1 to divide the swarm into k clusters
 - 3: Using Algorithm 3 to generate the new agent set
 - 4: **for** each new agent x_i in the new agent set **do**
 - 5: Compare x_i with the solution nearest to it in S and replace this agent if it is better than this agent
 - 6: **end for**
 - 7: Using Algorithm 4 to conduct local search
 - 8: Stop if the stopping criterion is met. Otherwise go to 2
-

fs is the species size bound in Algorithm 6. There is always an upper bound of the distance calculations which is $N(N-1)/2$. However, tighter upper bound can be obtained in the MMWOA.

In K-MMWOA, after obtaining the current centroids, the main operation is determining which cluster an agent should belong to. It requires $k(N-k)$ distance calculations in each generation. Since the loop bound is a certain number given by user and k is a number with a small bound, the computational complexity is $O(N)$. In FS-MMWOA, $N-1-(i-1)fs$ (where i is the loop variable and bound to the ceiling of N/fs) distance calculations are needed in each loop of Algorithm 2. Hence the total bound of distance calculations are $k(N-1) - k(k-1)fs/2 \approx (k+1)N/2 - k$. Therefore, both K-MMWOA and FS-MMWOA require $O(N)$ distance calculations.

In most real-world applications, the cost function evaluation is very time-consuming. In that case, the computational complexity of MMWOA can be estimated based on the number of cost function evaluations. In each generation, MMWOA generates N candidate positions which adds N times of cost function calculations. With a fixed generation bound, the computational complexity is $O(N)$. MMWOA randomly perform local search during the multimodal optimization process. Since the

Algorithm 6 FS-MMWOA

Input: the swarm size $sizeOfSwarm$, the species size bound fs

Output: the swarm with multiple best agents

- 1: Randomly initialize a swarm S with $sizeOfSwarm$ agents
 - 2: Randomly generate a number from fs
 - 3: Using Algorithm 2 to divide the swarm into a set of clusters
 - 4: Using Algorithm 3 to generate the new agent set
 - 5: **for** each new agent x_i in new agent set **do**
 - 6: Compare x_i with the agent nearest to it in S and update this agent if it is better than this agent
 - 7: **end for**
 - 8: Using Algorithm 4 to conduct local search
 - 9: Stop if the stopping criterion is met. Otherwise go to 2
-

sampling size is given as 4, the computational complexity of Gaussian sampling is less than $O(N)$. There is no cost function evaluation needed in the clustering process. Hence, the computational complexity of MMWOA is $O(N)$.

4. Numerical experiment

In this section, we shall use the multimodal benchmark functions and the evaluation criteria to test the performance of MMWOA and exhibit the results obtained. The details of multimodal benchmark functions are presented in subsection 4.1. The evaluation criteria are given in subsection 4.2. The experimental environment and the setting of experimental parameters are given in subsection 4.3. In subsection 4.4, the contour plots regarding the evolutionary progress of FS-MMWOA and K-MMWOA are drawn. Additionally, we employ MMWOA to solve a multimodal problem with nonlinear constraints. Corresponding results are depicted in subsection 4.5.

4.1. Multimodal benchmark functions

In this paper, 20 multimodal benchmark functions for multimodal function optimization in CEC'2013 are used to examine the performance of MMWOA proposed. All multimodal benchmark functions are formulated as maximization problems. The name, dimension, search range, peak value and global optimal number of these benchmark functions are listed in Table 1. Among these 20 functions, F1–F3 are 1D functions, and F4–F7 and F10–F13 are 2D functions. The results regarding F4–F7 and F10–F13 can be visualized in the form of 3D plot. Generally, the more dimension a function has, the more difficultly a multimodal optimizer can handle the problem.

4.2. Evaluation criteria

We introduce several widely used evaluate criteria to evaluate the performance of multimodal optimization algorithm over multiple runs. These evaluate criteria include peak ratio (PR), success rate (SR) and convergence speed (CS) [33]. The peak ratio can be formulated as follows:

$$PR = \frac{\sum_{run=1}^{NR} NPF_i}{NKP * NR}, \quad (4.1)$$

where NPF_i represents the number of global optima found in the i^{th} run, NKP is the number of known global optima, and NR denotes the number of runs.

The success rate can be calculated as follows:

$$SR = \frac{NSR}{NR} \quad (4.2)$$

where NSR is the number of successful runs. A successful run means all known global optima are found in this run.

The convergence speed can be calculated as follows:

$$CS = \frac{\sum_{run=1}^{NR} FE_i}{NR} \quad (4.3)$$

where FE_i is the number of function evaluations used in the i^{th} run.

As recommended in CEC'2013, all results are obtained over 30 independent runs. So the PR in results is the average percentage of all known global optima found for 30 runs. Likewise, SR and CS are the averages for 30 runs.

Table 1. Multimodal benchmark functions.

Func	Name	Dim	Range	F_{max}	No. of optima
F1	Five-Uneven-Peak Trap	1	[0, 30]	200.0	2
F2	Equal Maxima	1	[0, 1]	1.0	5
F3	Uneven Decreasing Maxima	1	[0, 1]	1.0	1
F4	Himmelblau	2	[-6, 6]	200.0	4
F5	Six-Hump Camel Back	2	$x \in [-1.9, 1.9],$ $y \in [-1.1, 1.1]$	1.0316	2
F6	Shubert(2D)	2	[-10, 10]	186.7309	18
F7	Vincent(2D)	2	[0.25, 10]	1.0	36
F8	Shubert(3D)	3	[-10, 10]	2709.0935	81
F9	Vincent(3D)	3	[0.25, 10]	1.0	216
F10	Modified Rastrigin	2	[0, 1]	-2.0	12
F11	Composition Function 1	2	[-5, 5]	0	6
F12	Composition Function 2	2	[-5, 5]	0	8
F13	Composition Function 3(2D)	2	[-5, 5]	0	6
F14	Composition Function 3(3D)	3	[-5, 5]	0	6
F15	Composition Function 4(3D)	3	[-5, 5]	0	8
F16	Composition Function 3(5D)	5	[-5, 5]	0	6
F17	Composition Function 4(5D)	5	[-5, 5]	0	8
F18	Composition Function 3(10D)	10	[-5, 5]	0	6
F19	Composition Function 4(10D)	10	[-5, 5]	0	8
F20	Composition Function 4(20D)	20	[-5, 5]	0	8

4.3. Experimental environment and parameter setting

All experiments are conducted on a server with four Intel Core i5–8300H 2.20 GHz CPUs and 8 GB memory. Programs are implemented with Python language and running on the Windows 64-bit systems. The parameters of MMWOA for 20 multimodal benchmark functions are shown in Table 2. “Func” is the function code; “SS” is the swarm size; “MaxFes” is the maximum number of evaluations of the function.

Table 2. Parameter setting.

Func	SS	MaxFes
F1–F5	80	5.0E+4
F6	100	2.0E+5
F7	300	2.0E+5
F8–F9	300	4.0E+5
F10	100	2.0E+5
F11–F13	200	2.0E+5
F14–F20	300	2.0E+5

The parameters which are not listed in Table 2 include: The number of sampling points is set to 4; The standard deviation of the Gaussian distribution is set to $1.0E-4$.

4.4. Algorithm comparison

Following the suggestion of CEC’2013, we compare PR, SR, and CS of K-MMWOA and FS-MMWOA with some state-of-the-art multimodal algorithms on 20 multimodal benchmark functions at accuracy level $1.0E+01$, $1.0E+02$, $1.0E+03$, $1.0E+04$, and $1.0E+05$, respectively. The PR, SR, and CS of various multimodal algorithms are shown in Tables 3–7. The best PR results are shown in bold for each algorithm, and bprs in each table represents the number of the best PR for a certain algorithm. More information regarding SDE and LIPS can be obtained from the supplementary material of [48].

It can be observed that PR of K-MMWOA or FS-MMWOA is 1 on F1–F3, F5 at all five accuracy level. Especially, FS-MMWOA and K-MMWOA outperform other optimizers on F1 at all accuracy level. Moreover, FS-MMWOA and K-MMWOA outperform other optimizers on F6, F9, F10, F14, F15, F17, and F18 at accuracy level of $1.0E+01$, and outperform other optimizers on F7, F9, F10 and F18 at accuracy level of $1.0E+02$. As precision increases, FS-MMWOA and K-MMWOA still outperform other algorithms on some functions.

The global optimal number of F9 is 216. It is hard for other optimizers to find multiple optimums, while in contrast, FS-MMWOA and K-MMWOA work well. For the benchmark functions in very high dimension such as F19 (10D) and F20 (20D), FS-MMWOA and K-MMWOA lost the search ability, same as other multimodal optimizers.

Although both FS-MMWOA and K-MMWOA show competitive multimodal search ability on the 20 benchmark functions, K-MMWOA show better performance than FS-MMWOA for most cases. So we can make a judgement that the clustering method plays an important role in MMWOA.

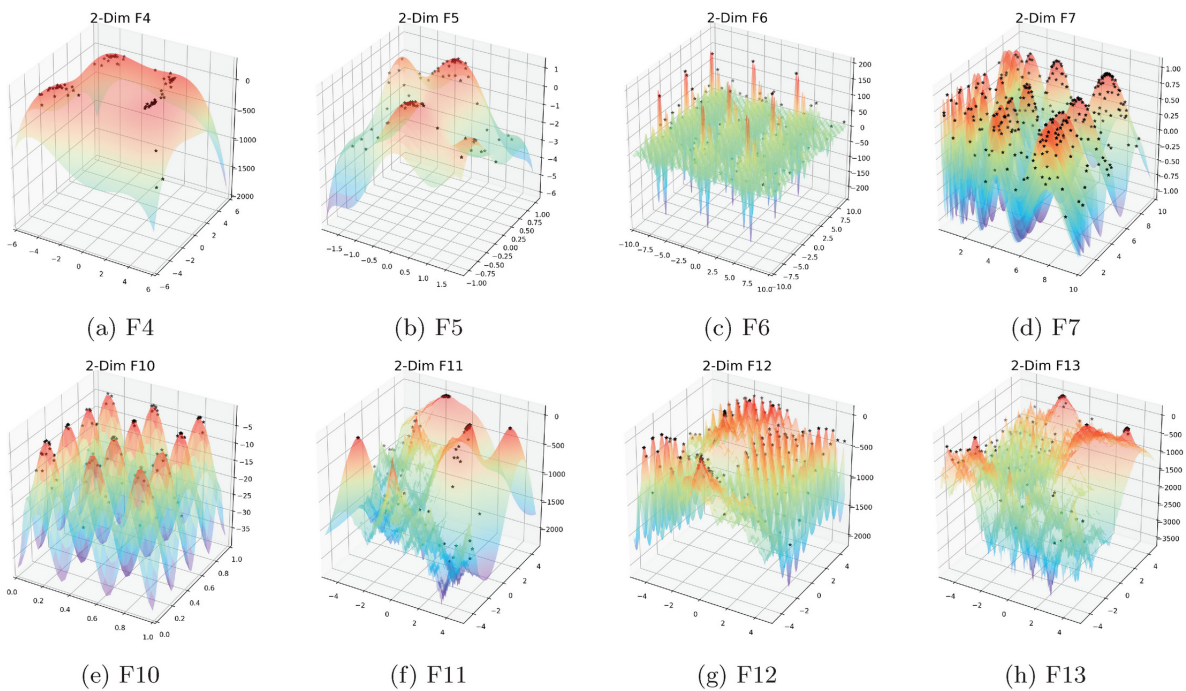


Figure 2. Final distribution of agents on some benchmark landscapes (FS-MMWOA).

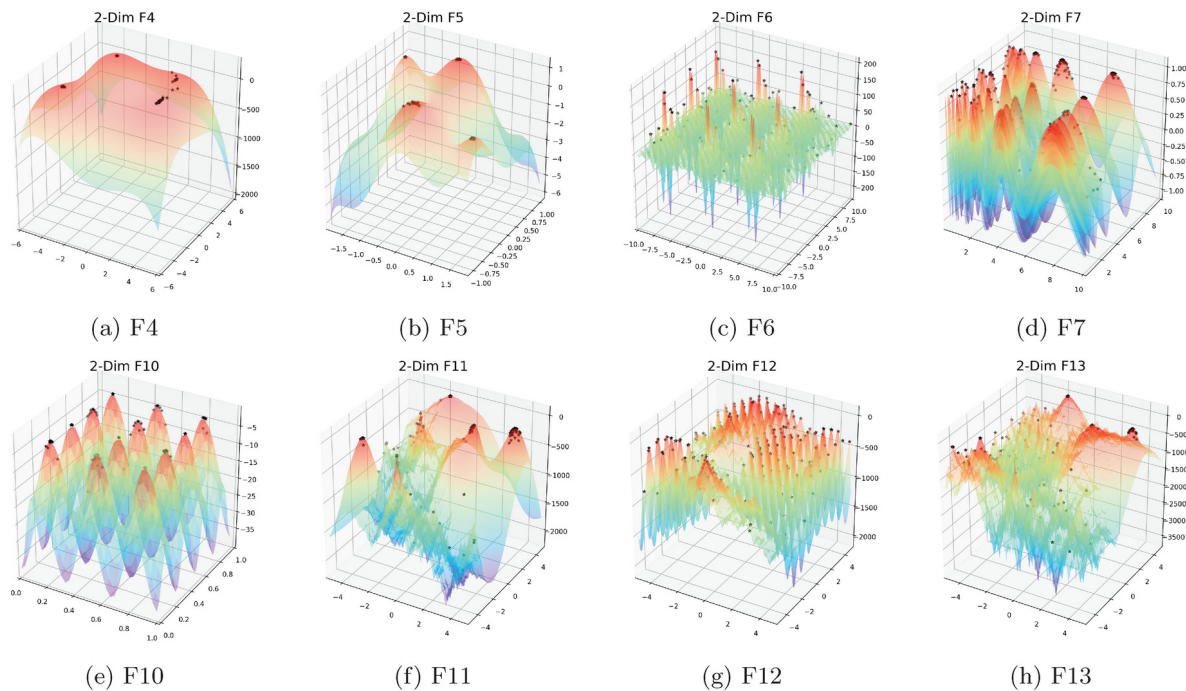


Figure 3. Final distribution of agents on some benchmark landscapes (K-MMWOA).

Figures 2 and 3 exhibit the multimodal search dynamics of the FS-MMWOA and K-MMWOA. The 3D plots are suitable for visualization of MMWOA on 2D optimization functions. So only the results of F4–F7 and F10–F13 are shown here.

Additionally, we draw the contour plots of FS-MMWOA and K-MMWOA on F4–F7 and F10–F13.

To keep the paper concise, we put the contour plots to the Appendix A. The contour plots provide more details of the evolutionary progress, see Figures 4 and 5.

As shown in Figures 4 and 5, from left columns to right columns, the multimodal search process can be divided into two stage: the exploration stage (early search stage) and the exploitation stage (late search stage) .

During the exploration stage, we take advantage of the basic WOA to obtain the high-quality candidate positions for the next generation. Just like encircling prey in the WOA, at each iteration, search agents update their positions with respect to either a randomly chosen position or the optimum obtained so far. Unlike the basic WOA, MMWOA divides the swarm into different species. Thus the best solution which was being encircled in WOA becomes several best solutions in different clusters. Another enhancement for exploration is the updating mechanism in Algorithm 5 and Algorithm 6. Since MMWOA always update the position of current agent to the nearest candidate position, it promotes diversity to a certain extent.

During the exploitation stage, MMWOA take advantage of the shrinking encircling and bubble-net attacking mechanism in WOA to find better optimum near the current optimum. In addition, the exploitation ability of MMWOA is enhanced by the Gaussian sampling in local search. The sampling points near the current local optimum are likely to be better than current optimum.

Agents in different clusters are labelled in different colours in Figures 4 and 5. MMWOA performs the clustering operation in each generation or every certain number of generations. In the FS-MMWOA, due to the eliminating effect, the agents within a same cluster scatter in a larger range. In contrast, for K-MMWOA, agents in the same cluster gather more together. Therefore, FS-MMWOA has better exploration ability, while K-MMWOA has better exploitation ability.

As shown in Table 3, when the accuracy level is low, FS-MMWOA outperforms K-MMWOA because of its better exploration ability. Multimodal search for high-dimension problem needs better exploration ability, so FS-MMWOA shows good performance for high-dimension problem, such as for F19 in Table 4. When the accuracy level is high, better exploitation ability is needed. That is why K-MMWOA shows massive advantage over FS-MMWOA in Tables 5–7.

4.5. Addressing multimodal problem with nonlinear constraints

Next we shall employ MMWOA to address a multimodal problem with non-linear constraints. The CMMP(n,G,L) problem is presented by Deb in [42], which has n variables, L number of local optima, and G number of optima, where G is equal to 2^J and J is the number of constraints.

The CMMP(n,G,L) problem can be formulated as follows:

$$\begin{aligned} \min. f(x) &= \sum_{i=1}^n x_i^2, & (4.4) \\ \text{s.t. } g_1(x) &= x_1^2 + 4x_2^2 + 9x_3^2 + \cdots + n^2x_n^2 \geq n^2, \\ g_2(x) &= n^2x_1^2 + x_2^2 + 4x_3^2 + \cdots + (n-1)^2x_n^2 \geq n^2, \\ &\vdots \\ g_J(x) &= C_{J,1}^2x_1^2 + C_{J,2}^2x_2^2 + C_{J,3}^2x_3^2 + \cdots + C_{J,n}^2x_n^2 \geq n^2, \\ &-(n+1) \leq x_i \leq (n+1), \text{ for } i = 1, 2, \dots, n, \end{aligned}$$

where



Figure 4. Evolutionary progress of FS-MMWOA on F4–F7 and F10–F13.



Figure 5. Evolutionary progress of K-MMWOA on F4–F7 and F10–F13.

Table 3. Comparison of PR,SR, and CS among FS-MMWOA, K-MMWOA, and some state-of-the-art multimodal methods on 20 benchmark functions at accuracy level $\epsilon=1.0E-01$.

1.0E-01												
F	SDE			LIPS			FS-MMWOA			K-MMWOA		
	PR	SR	CS	PR	SR	CS	PR	SR	CS	PR	SR	CS
F1	0.657	0.373	3.24E+4	0.833	0.686	1.64E+4	1.000	1.000	2.69E+2	1.000	1.000	1.70E+2
F2	1.000	1.000	1.05E+2	1.000	1.000	1.22E+2	1.000	1.000	2.68E+2	1.000	1.000	1.75E+2
F3	1.000	1.000	8.74E+1	1.000	1.000	8.63E+1	1.000	1.000	2.64E+2	1.000	1.000	1.69E+2
F4	1.000	1.000	2.43E+3	1.000	1.000	8.89E+2	1.000	1.000	1.19E+4	1.000	1.000	3.38E+3
F5	1.000	1.000	1.90E+2	1.000	1.000	2.54E+2	1.000	1.000	8.77E+2	1.000	1.000	3.19E+2
F6	0.056	0.000	2.00E+5	0.276	0.000	2.00E+5	0.500	0.000	2.00E+5	0.333	0.000	2.00E+5
F7	1.000	1.000	1.35E+3	0.976	0.961	9.44E+3	1.000	1.000	8.48E+3	1.000	1.000	9.48E+3
F8	0.015	0.000	4.00E+5	0.086	0.000	4.00E+5	0.000	0.000	4.00E+5	0.049	0.000	4.00E+5
F9	0.011	0.000	4.00E+5	0.105	0.000	4.00E+5	0.588	0.000	4.00E+5	0.491	0.000	3.51E+5
F10	0.229	0.098	1.81E+5	0.781	0.216	1.57E+5	0.917	0.803	1.63E+5	1.000	1.000	2.57E+4
F11	0.944	0.922	2.15E+4	1.000	1.000	4.09E+3	1.000	1.000	4.82E+5	1.000	1.000	1.63E+4
F12	0.208	0.000	2.00E+5	0.596	0.000	2.00E+5	0.375	0.000	2.00E+5	0.500	0.000	2.00E+5
F13	0.618	0.471	1.09E+5	0.941	0.745	5.49E+4	1.000	1.000	7.00E+4	0.667	0.333	1.56E+5
F14	0.938	0.922	4.13E+4	0.990	0.980	1.39E+4	1.000	1.000	3.35E+5	1.000	1.000	3.27E+5
F15	0.581	0.549	1.87E+5	0.537	0.333	2.70E+5	1.000	1.000	2.21E+5	1.000	1.000	1.77E+5
F16	0.647	0.647	1.72E+5	1.000	1.000	8.74E+3	1.000	1.000	1.50E+5	1.000	1.000	6.86E+4
F17	0.574	0.569	2.02E+5	0.257	0.118	3.54E+5	1.000	1.000	3.97E+5	1.000	1.000	1.59E+5
F18	0.157	0.157	3.69E+5	0.725	0.725	1.20E+5	1.000	1.000	1.11E+5	1.000	1.000	4.06E+4
F19	0.145	0.039	3.93E+5	0.010	0.000	4.00E+5	0.125	0.000	4.00E+5	0.000	0.000	4.00E+5
F20	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
bprs	6			8			15			13		

Table 4. Comparison of PR,SR, and CS among FS-MMWOA, K-MMWOA, and some state-of-the-art multimodal methods on 20 benchmark functions at accuracy level $\epsilon=1.0E-02$.

1.0E-02												
F	SDE			LIPS			FS-MMWOA			K-MMWOA		
	PR	SR	CS	PR	SR	CS	PR	SR	CS	PR	SR	CS
F1	0.657	0.373	3.24E+4	0.833	0.686	1.64E+4	1.000	1.000	2.69E+2	1.000	1.000	1.70E+2
F2	1.000	1.000	2.01E+2	1.000	1.000	3.04E+2	1.000	1.000	9.02E+2	1.000	1.000	4.50E+2
F3	1.000	1.000	2.58E+2	1.000	1.000	2.60E+2	1.000	1.000	4.72E+2	1.000	1.000	2.94E+2
F4	1.000	1.000	2.78E+3	1.000	1.000	1.49E+3	1.000	1.000	2.69E+4	1.000	1.000	7.02E+3
F5	1.000	1.000	7.18E+2	1.000	1.000	5.07E+2	1.000	1.000	1.71E+3	1.000	1.000	1.23E+3
F6	0.056	0.000	2.00E+5	0.257	0.000	2.00E+5	0.222	0.000	2.00E+5	0.222	0.000	2.00E+5
F7	0.054	0.000	2.00E+5	0.400	0.000	2.00E+5	0.722	0.621	8.48E+4	0.694	0.000	2.00E+5
F8	0.015	0.000	4.00E+5	0.085	0.000	4.00E+5	0.000	0.000	4.00E+5	0.037	0.000	4.00E+5
F9	0.011	0.000	4.00E+5	0.105	0.000	4.00E+5	0.222	0.000	4.00E+5	0.227	0.000	4.00E+5
F10	0.147	0.000	2.00E+5	0.758	0.020	1.96E+5	0.833	0.692	1.86E+5	0.917	0.000	2.00E+5
F11	0.314	0.000	2.00E+5	0.974	0.843	5.33E+4	0.500	0.000	2.00E+5	0.500	0.000	2.00E+5
F12	0.208	0.000	2.00E+5	0.583	0.000	2.00E+5	0.375	0.000	2.00E+5	0.375	0.000	2.00E+5
F13	0.297	0.000	2.00E+5	0.804	0.176	1.70E+5	0.500	0.000	2.00E+5	0.500	0.000	2.00E+5
F14	0.216	0.000	4.00E+5	0.644	0.000	4.00E+5	0.167	0.000	4.00E+5	0.667	0.000	4.00E+5
F15	0.108	0.000	4.00E+5	0.336	0.000	4.00E+5	0.125	0.000	4.00E+5	0.417	0.000	4.00E+5
F16	0.108	0.000	4.00E+5	0.307	0.000	4.00E+5	0.167	0.000	4.00E+5	0.500	0.000	4.00E+5
F17	0.076	0.000	4.00E+5	0.162	0.000	4.00E+5	0.000	0.000	4.00E+5	0.250	0.000	4.00E+5
F18	0.026	0.000	4.00E+5	0.121	0.000	4.00E+5	0.167	0.000	4.00E+5	0.167	0.000	4.00E+5
F19	0.110	0.000	4.00E+5	0.005	0.000	4.00E+5	0.125	0.000	4.00E+5	0.000	0.000	4.00E+5
F20	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
bprs	4			9			8			12		

Table 5. Comparison of PR,SR, and CS among FS-MMWOA, K-MMWOA, and some state-of-the-art multimodal methods on 20 benchmark functions at accuracy level $\epsilon=1.0E-03$.

1.0E-03												
F	SDE			LIPS			FS-MMWOA			K-MMWOA		
	PR	SR	CS	PR	SR	CS	PR	SR	CS	PR	SR	CS
F1	0.657	0.373	3.24E+4	0.833	0.686	1.64E+4	1.000	1.000	2.69E+2	1.000	1.000	1.70E+2
F2	0.976	0.961	2.35E+3	1.000	1.000	4.93E+2	1.000	1.000	2.57E+3	1.000	1.000	1.31E+3
F3	1.000	1.000	4.68E+2	0.980	0.980	1.40E+3	1.000	1.000	7.16E+2	1.000	1.000	4.26E+2
F4	0.284	0.000	5.00E+4	0.990	0.961	4.28E+3	0.500	0.000	5.00E+4	0.808	0.4	3.61E+4
F5	0.971	0.941	4.09E+3	1.000	1.000	9.35E+2	1.000	1.000	2.61E+3	1.000	1.000	2.76E+3
F6	0.056	0.000	2.00E+5	0.252	0.000	2.00E+5	0.056	0.000	2.00E+5	0.222	0.000	2.00E+5
F7	0.054	0.000	2.00E+5	0.400	0.000	2.00E+5	0.500	0.000	2.00E+5	0.528	0.000	2.00E+5
F8	0.015	0.000	4.00E+5	0.084	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
F9	0.011	0.000	4.00E+5	0.104	0.000	4.00E+5	0.102	0.000	4.00E+5	0.116	0.000	4.00E+5
F10	0.147	0.000	2.00E+5	0.757	0.000	2.00E+5	0.750	0.000	2.00E+5	0.667	0.000	2.00E+5
F11	0.314	0.000	2.00E+5	0.974	0.843	5.74E+4	0.500	0.000	2.00E+5	0.500	0.000	2.00E+5
F12	0.208	0.000	2.00E+5	0.578	0.000	2.00E+5	0.250	0.000	2.00E+5	0.250	0.000	2.00E+5
F13	0.297	0.000	2.00E+5	0.797	0.176	1.71E+5	0.333	0.000	2.00E+5	0.500	0.000	2.00E+5
F14	0.216	0.000	4.00E+5	0.644	0.000	4.00E+5	0.167	0.000	4.00E+5	0.667	0.000	4.00E+5
F15	0.108	0.000	4.00E+5	0.336	0.000	4.00E+5	0.125	0.000	4.00E+5	0.417	0.000	4.00E+5
F16	0.108	0.000	4.00E+5	0.307	0.000	4.00E+5	0.167	0.000	4.00E+5	0.500	0.000	4.00E+5
F17	0.076	0.000	4.00E+5	0.162	0.000	4.00E+5	0.000	0.000	4.00E+5	0.250	0.000	4.00E+5
F18	0.026	0.000	4.00E+5	0.111	0.000	4.00E+5	0.167	0.000	4.00E+5	0.167	0.000	4.00E+5
F19	0.108	0.000	4.00E+5	0.002	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
F20	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
bprs	2			8			5			11		

Table 6. Comparison of PR,SR, and CS among FS-MMWOA, K-MMWOA, and some state-of-the-art multimodal methods on 20 benchmark functions at accuracy level $\epsilon=1.0E-04$.

1.0E-04												
F	SDE			LIPS			FS-MMWOA			K-MMWOA		
	PR	SR	CS	PR	SR	CS	PR	SR	CS	PR	SR	CS
F1	0.657	0.373	3.24E+4	0.833	0.686	1.64E+4	1.000	1.000	2.69E+2	1.000	1.000	1.70E+2
F2	0.737	0.529	2.39E+4	1.000	1.000	7.61E+2	1.000	1.000	4.23E+3	1.000	1.000	3.12E+3
F3	1.000	1.000	6.82E+2	0.961	0.961	2.66E+3	1.000	1.000	1.16E+3	1.000	1.000	7.62E+2
F4	0.284	0.000	5.00E+4	0.990	0.961	5.00E+3	0.500	0.000	5.00E+4	0.517	0.067	4.74E+4
F5	0.922	0.843	9.25E+3	1.000	1.000	1.49E+3	1.000	1.000	5.27E+3	1.000	1.000	4.86E+3
F6	0.056	0.000	2.00E+5	0.246	0.000	2.00E+5	0.056	0.000	2.00E+5	0.056	0.000	2.00E+5
F7	0.054	0.000	2.00E+5	0.400	0.000	2.00E+5	0.278	0.000	2.00E+5	0.361	0.000	2.00E+5
F8	0.015	0.000	4.00E+5	0.084	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
F9	0.011	0.000	4.00E+5	0.104	0.000	4.00E+5	0.037	0.000	4.00E+5	0.079	0.000	4.00E+5
F10	0.147	0.000	2.00E+5	0.748	0.000	2.00E+5	0.583	0.000	2.00E+5	0.583	0.000	2.00E+5
F11	0.314	0.000	2.00E+5	0.974	0.843	5.90E+4	0.500	0.000	2.00E+5	0.500	0.000	2.00E+5
F12	0.208	0.000	2.00E+5	0.574	0.000	2.00E+5	0.000	0.000	2.00E+5	0.250	0.000	2.00E+5
F13	0.297	0.000	2.00E+5	0.794	0.176	1.72E+5	0.333	0.000	2.00E+5	0.389	0.000	2.00E+5
F14	0.216	0.000	4.00E+5	0.644	0.000	4.00E+5	0.167	0.000	4.00E+5	0.556	0.000	4.00E+5
F15	0.108	0.000	4.00E+5	0.336	0.000	4.00E+5	0.125	0.000	4.00E+5	0.250	0.000	4.00E+5
F16	0.108	0.000	4.00E+5	0.304	0.000	4.00E+5	0.167	0.000	4.00E+5	0.333	0.000	4.00E+5
F17	0.076	0.000	4.00E+5	0.162	0.000	4.00E+5	0.000	0.000	4.00E+5	0.125	0.000	4.00E+5
F18	0.026	0.000	4.00E+5	0.098	0.000	4.00E+5	0.167	0.000	4.00E+5	0.167	0.000	4.00E+5
F19	0.105	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
F20	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
bprs	2			13			5			6		

Table 7. Comparison of PR,SR, and CS among FS-MMWOA, K-MMWOA, and some state-of-the-art multimodal methods on 20 benchmark functions at accuracy level $\epsilon=1.0E-05$.

1.0E-05												
F	SDE			LIPS			FS-MMWOA			K-MMWOA		
	PR	SR	CS	PR	SR	CS	PR	SR	CS	PR	SR	CS
F1	0.657	0.373	3.24E+4	0.833	0.686	1.64E+4	1.000	1.000	2.69E+2	1.000	1.000	1.70E+2
F2	0.584	0.275	3.65E+4	1.000	1.000	1.36E+3	1.000	1.000	1.20E+4	1.000	1.000	3.12E+3
F3	1.000	1.000	8.33E+2	0.961	0.961	3.04E+3	1.000	1.000	1.82E+3	1.000	1.000	1.36E+3
F4	0.284	0.000	5.00E+4	0.990	0.961	5.66E+3	0.250	0.000	5.00E+4	0.258	0.033	4.88E+4
F5	0.853	0.706	1.62E+4	1.000	1.000	2.18E+3	1.000	1.000	7.37E+3	1.000	1.000	7.08E+3
F6	0.056	0.000	2.00E+5	0.244	0.000	2.00E+5	0.000	0.000	2.00E+5	0.000	0.000	2.00E+5
F7	0.054	0.000	2.00E+5	0.397	0.000	2.00E+5	0.194	0.000	2.00E+5	0.222	0.000	2.00E+5
F8	0.015	0.000	4.00E+5	0.084	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
F9	0.011	0.000	4.00E+5	0.103	0.000	4.00E+5	0.014	0.000	4.00E+5	0.037	0.000	4.00E+5
F10	0.147	0.000	2.00E+5	0.747	0.000	2.00E+5	0.333	0.000	2.00E+5	0.111	0.500	2.00E+5
F11	0.314	0.000	2.00E+5	0.974	0.843	6.10E+4	0.333	0.000	2.00E+5	0.500	0.000	2.00E+5
F12	0.208	0.000	2.00E+5	0.574	0.000	2.00E+5	0.000	0.000	2.00E+5	0.250	0.000	2.00E+5
F13	0.297	0.000	2.00E+5	0.794	0.176	1.72E+5	0.333	0.000	2.00E+5	0.389	0.000	2.00E+5
F14	0.216	0.000	4.00E+5	0.644	0.000	4.00E+5	0.167	0.000	4.00E+5	0.556	0.000	4.00E+5
F15	0.108	0.000	4.00E+5	0.336	0.000	4.00E+5	0.125	0.000	4.00E+5	0.125	0.000	4.00E+5
F16	0.108	0.000	4.00E+5	0.304	0.000	4.00E+5	0.167	0.000	4.00E+5	0.333	0.000	4.00E+5
F17	0.076	0.000	4.00E+5	0.162	0.000	4.00E+5	0.000	0.000	4.00E+5	0.125	0.000	4.00E+5
F18	0.026	0.000	4.00E+5	0.095	0.000	4.00E+5	0.167	0.000	4.00E+5	0.167	0.000	4.00E+5
F19	0.105	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
F20	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5	0.000	0.000	4.00E+5
bprs	2			14			5			6		

$$C_{j,k} = \begin{cases} (n - j + k + 1) \bmod n, & \text{if } (n - j + k + 1) \bmod n \neq 0 \\ n, & \text{otherwise.} \end{cases}$$

In the paper, we use the objective function $f(x_1, x_2) = x_1^2 + (x_2 - 0.2)^2$ (CMMP(2,2,2)), which has two global optimums: $(\sqrt{0.8}, \sqrt{0.8})$, $(-\sqrt{0.8}, \sqrt{0.8})$ with $f(x_1, x_2) = 1.282$ and two local optimums: $(-\sqrt{0.8}, -\sqrt{0.8})$, $(\sqrt{0.8}, -\sqrt{0.8})$ with $f(x_1, x_2) = 1.998$.

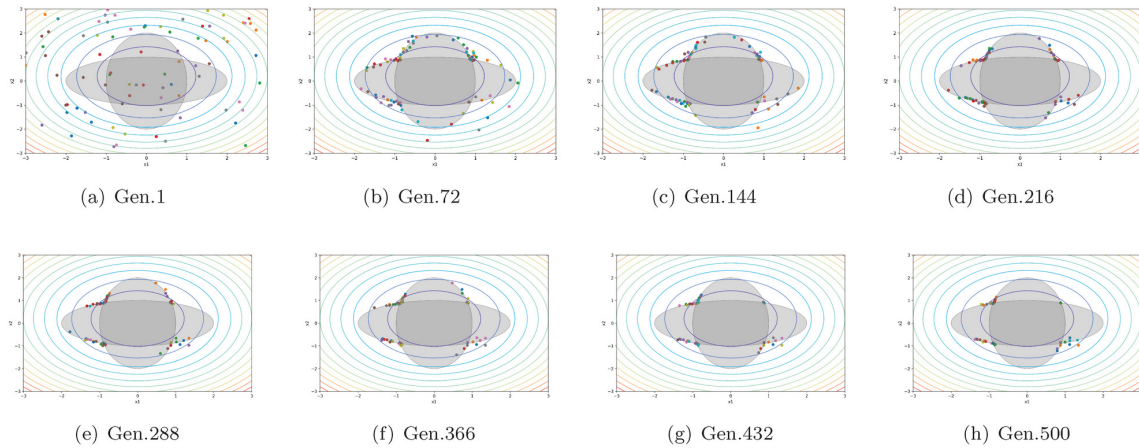


Figure 6. Evolutionary progress of FS-MMWOA for the CMMP(2,2,2) problem.

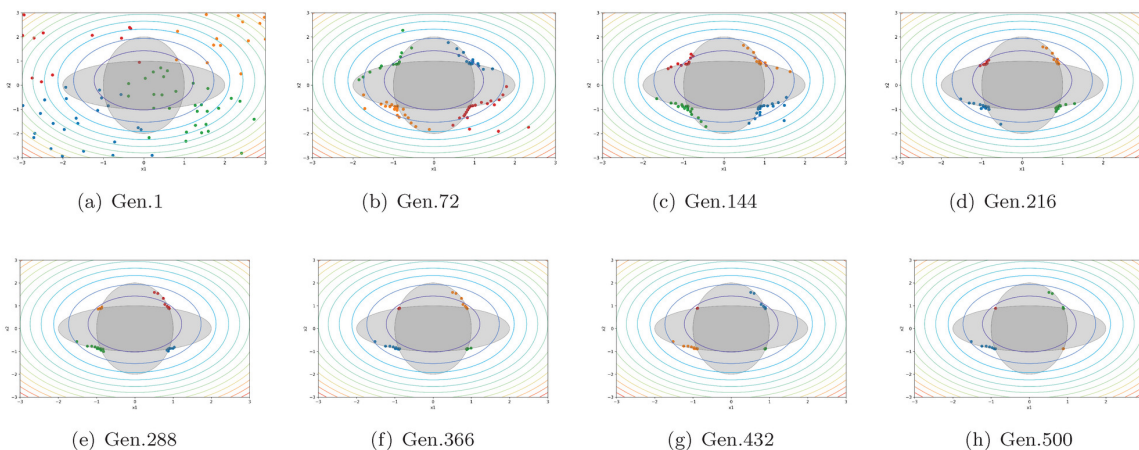


Figure 7. Evolutionary progress of K-MMWOA for the CMMP(2,2,2) problem.

By transforming the non-linear constraints to the penalty terms, we solve the CMMP problem using MMWOA. The swarm size is set to 80. The number of iterations is limited to less than 500. The evolutionary progress of FS-MMWOA and K-MMWOA are illustrated in the contour plots of Figures 6 and 7. Both FS-MMWOA and K-MMWOA can find all the optimums. K-MMWOA needs less iterations to achieve the goal.

5. Conclusion

In this paper, we proposed a multimodal optimization algorithm based on WOA which is a popular nature inspired metaheuristics. The niching techniques including K-means clustering and the clustering with a fixed cluster size bound were introduced to the proposed algorithm to enhance its multimodal search ability. In addition, the local search technique was absorbed in MMWOA to improve the local search ability of the best agent of each species. We tested our proposed algorithm on 20 multimodal benchmark functions and a CMMP problem with non-linear constraints. The experimental results verified that MMWOA has competitive ability in solving multimodal optimization problems.

Since the MMWOA performs better at the low accuracy levels than at the high accuracy levels. One future work is further improving the local search ability of MMWOA. Another future work is using the MMWOA to solve more multimodal optimization problems in the real-life applications.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants of the general technical foundation research joint fund (Project No. U1636208). This article does not contain any studies with human participants or animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

Conflict of interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. J. Kennedy, Particle Swarm Optimization, in *Encyclopedia of Machine Learning* (eds. C. Sammut and G.I. Webb), Springer US, (2010), 760–766.
2. X. Yang and D. Suash, *Cuckoo Search via Lévy flights*, 2009 World Congress on Nature & Biologically Inspired Computing, (2009), 210–214. Available from: <https://ieeexplore.ieee.org/abstract/document/5393690/>.
3. X. S. Yang, *Firefly Algorithms for Multimodal Optimization*, International symposium on stochastic algorithms, (2009), 169–178. Available from: https://link.springer.com/chapter/10.1007/978-3-642-04944-6_14.
4. S. Mirjalili and A. Lewis, The Whale Optimization Algorithm, *Adv. Eng. Software*, **95** (2016), 51–67.
5. M. A. E. Aziz, A. A. Ewees and A. E. Hassanien, Multi-objective whale optimization algorithm for content-based image retrieval, *Multimedia Tools Appl.*, **77** (2018), 26135–26172.
6. A. N. Jadhav and N. Gomathi, WGC: Hybridization of exponential grey wolf optimizer with whale optimization for data clustering, *Alexandria Eng. J.*, **57** (2018), 1569–1584.

7. M. K. Hassan, A. I. El Desouky, S. M. Elghamrawy, et al., A Hybrid Real-time remote monitoring framework with NB-WOA algorithm for patients with chronic diseases, *Future Gener. Comput. Syst.*, **93** (2019), 77–95.
8. A. Mostafa, A. E. Hassanien, M. Houseni, et al., Liver segmentation in MRI images based on whale optimization algorithm, *Multimedia Tools Appl.*, **76** (2017), 24931–24954.
9. G. Hassan and A. E. Hassanien, Retinal fundus vasculature multilevel segmentation using whale optimization algorithm, *Signal Image Video Process.*, **12** (2018), 263–270.
10. Y. Miao, M. Zhao, V. Makis, et al., Optimal swarm decomposition with whale optimization algorithm for weak feature extraction from multicomponent modulation signal, *Mech. Syst. Signal Process.*, **122** (2019), 673–691.
11. X. Zhang, J. Zhao, X. Zhang, et al., A novel hybrid compound fault pattern identification method for gearbox based on NIC, MFDFA and WOASVM, *J. Mech. Sci. Technol.*, **33** (2019), 1097–1113.
12. D. Oliva, M. Abd El Aziz and A. Ella Hassanien, Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm, *Appl. Energy*, **200** (2017), 141–154.
13. O. S. Elazab, H. M. Hasanien, M. A. Elgendy, et al., Parameters estimation of single- and multiple-diode photovoltaic model using whale optimisation algorithm, *IET Renewable Power Gener.*, **12** (2018), 1755–1761.
14. G. Ren, R. Yang, R. Yang, et al., A parameter estimation method for fractional-order nonlinear systems based on improved whale optimization algorithm, *Mod. Phys. Lett. B*, **33** (2019), 1950075.
15. K. b. o. Medani, S. Sayah and A. Bekrar, Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system, *Electr. Power Syst. Res.*, **163** (2018), 696–705.
16. K. S. Simhadri, B. Mohanty and S. K. Panda, Comparative performance analysis of 2DOF state feedback controller for automatic generation control using whale optimization algorithm, *Optim. Control Appl. Methods*, **40** (2019), 24–42.
17. D. Yousri, D. Allam and M. B. Eteiba, Chaotic whale optimizer variants for parameters estimation of the chaotic behaviour in Permanent Magnet Synchronous Motor, *Appl. Soft Comput.*, **74** (2019), 479–503.
18. M. Abdel-Basset, D. El-Shahat and A. K. Sangaiah, A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem, *Int. J. Mach. Learn. Cybern.*, **10** (2019), 495–514.
19. J. Ghahremani-Nahr, R. Kian and E. Sabet, A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm, *Expert Syst. Appl.*, **116** (2019), 454–471.
20. L. Wang, W. H. Wu, J. Y. Qi, et al., Wireless Sensor Network Coverage Optimization based on Whale Group Algorithm, *Comput. Sci. Inf. Syst.*, **15** (2018), 569–583.
21. I. Aljarah, H. Faris and S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, *Soft Comput.*, **22** (2018), 1–15.
22. M. Abdel-Basset, D. El-Shahat, I. El-henawy, et al., A Novel Whale Optimization Algorithm for Cryptanalysis in Merkle-Hellman Cryptosystem, *Mobile Networks Appl.*, **23** (2018), 723–733.

23. M. A. M. Majeed, A hybrid of WOA and mGWO algorithms for global optimization and analog circuit design automation, *COMPEL - Int. J. Comput. Math. Electr. Electron. Eng.*, **38** (2019), 452–476.
24. V. K. Bohat and K. V. Arya, A new heuristic for multilevel thresholding of images, *Expert Syst. Appl.*, **117** (2019), 176–203.
25. H. Li, J. Zhang and J. Yi, Computational source term estimation of the Gaussian puff dispersion, *Soft Comput.*, **23** (2019), 59–75.
26. Y. Chen, R. Vepa and M. H. Shaheed, Enhanced and speedy energy extraction from a scaled-up pressure retarded osmosis process with a whale optimization based maximum power point tracking, *Energy*, **153** (2018), 618–627.
27. M. Abdel-Basset, G. Manogaran, D. El-Shahat, et al., Integrating the whale algorithm with Tabu search for quadratic assignment problem: A new approach for locating hospital departments, *Appl. Soft Comput.*, **73** (2018), 530–546.
28. I. G. Rajathi and W. G. Jiji, Chronic Liver Disease Classification Using Hybrid Whale Optimization with Simulated Annealing and Ensemble Classifier, *Symmetry*, **11** (2019), 33.
29. M. M. Mafarja and S. Mirjalili, Hybrid Whale Optimization Algorithm with simulated annealing for feature selection, *Neurocomputing*, **260** (2017), 302–312.
30. M. Bhowmik and P. Malathi, Spectrum Sensing in Cognitive Radio Using Actor–Critic Neural Network with Krill Herd-Whale Optimization Algorithm, *Wireless Pers. Commun.*, **105** (2019), 335–354.
31. E. Emary, H. M. Zawbaa and M. Sharawi, Impact of Lévy flight on modern meta-heuristic optimizers, *Appl. Soft Comput.*, **75** (2019), 775–789.
32. Y. Khalil, M. Alshayegi and I. Ahmad, Distributed Whale Optimization Algorithm based on MapReduce, *Concurr. Comp. Pract. E.*, **31** (2019), e4872.
33. X. Li, M. G. Epitropakis, K. Deb, et al., Seeking Multiple Solutions: An Updated Survey on Niching Methods and Their Applications, *IEEE Trans. Evol. Comput.*, **21** (2017), 518–538.
34. B. Sareni and L. Krahenbuhl, Fitness sharing and niching methods revisited, *IEEE Trans. Evol. Comput.*, **2** (1998), 97–106.
35. J. E. Fieldsend, *Running Up Those Hills: Multi-modal search with the niching migratory multi-swarm optimiser*, 2014 IEEE Congress on Evolutionary Computation, (2014), 2593–2600. Available from: <https://ieeexplore.ieee.org/abstract/document/6900309>.
36. R. Brits, A. P. Engelbrecht and F. van den Bergh, Locating multiple optima using particle swarm optimization, *Appl. Math. Comput.*, **189** (2007), 1859–1883.
37. B. Y. Qu, P. N. Suganthan and J. J. Liang, Differential Evolution With Neighborhood Mutation for Multimodal Optimization, *IEEE Trans. Evol. Comput.*, **16** (2012), 601–614.
38. N. Nekouie and M. Yaghoobi, A new method in multimodal optimization based on firefly algorithm, *Artif. Intell. Rev.*, **46** (2016), 267–287.
39. H. Banati and R. Chaudhary, Multi-Modal Bat Algorithm with Improved Search (MMBAIS), *J. Comput. Sci.*, **23** (2017), 130–144.

40. G. Jorge, C. Erik and A. Omar, Flower Pollination Algorithm for Multimodal Optimization, *Int. J. Comput. Intell. Syst.*, **10** (2017), 627–646.
41. D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82.
42. K. Deb and A. Saha, *Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach*, Proceedings of the 12th annual conference on Genetic and evolutionary computation, (2010), 447–454. Available from: <https://dl.acm.org/citation.cfm?id=1830568>.
43. Z. Michalewicz and M. Schoenauer, Evolutionary Algorithms for Constrained Parameter Optimization Problems, *Evol. Comput.*, **4** (1996), 1–32.
44. X. Li, *Efficient differential evolution using speciation for multimodal function optimization*, Proceedings of the 7th annual conference on Genetic and evolutionary computation, (2005), 873–880. Available from: <https://dl.acm.org/citation.cfm?id=1068156>.
45. A. Petrowski, *A clearing procedure as a niching method for genetic algorithms*, Proceedings of IEEE International Conference on Evolutionary Computation, (1996), 798–803. Available from: <https://ieeexplore.ieee.org/abstract/document/542703>.
46. R. Thomsen, *Multimodal optimization using crowding-based differential evolution*, Proceedings of the 2004 Congress on Evolutionary Computation, (2004), 1382–1389. Available from: <https://ieeexplore.ieee.org/abstract/document/1331058>.
47. B. Y. Qu, P. N. Suganthan and S. Das, A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization, *IEEE Trans. Evol. Comput.*, **17** (2013), 387–402.
48. Q. Yang, W. Chen, Z. Yu, et al., Adaptive Multimodal Continuous Ant Colony Optimization, *IEEE Trans. Evol. Comput.*, **21** (2017), 191–205.
49. J. A. Goldbogen, A. S. Friedlaender, J. Calambokidis, et al., Integrative Approaches to the Study of Baleen Whale Diving Behavior, Feeding Performance, and Foraging Ecology, *BioScience*, **63** (2013), 90–100.
50. W. A. Watkins and W. E. Schevill, Aerial Observation of Feeding Behavior in Four Baleen Whales: *Eubalaena glacialis*, *Balaenoptera borealis*, *Megaptera novaeangliae*, and *Balaenoptera physalus*, *J. Mammal.*, **60** (1979), 155–163.
51. S. Weiguo, S. Swift, Z. Leishi, et al., A weighted sum validity function for clustering with a hybrid niching genetic algorithm, *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)*, **35** (2005), 1156–1167.
52. H. Li and J. Zhang, Fast source term estimation using the PGA-NM hybrid method, *Eng. Appl. Artif. Intell.*, **62** (2017), 68–79.



AIMS Press

© 2020 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)