Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04015-0

ORIGINAL ARTICLE

An enhanced associative learning-based exploratory whale optimizer for global optimization

Ali Asghar Heidari[1] · Ibrahim Aljarah[2] · Hossam Faris[2] · Huiling Chen[3] · Jie Luo[3] · Seyedali Mirjalili[4]

Received: 5 July 2018 / Accepted: 8 January 2019

Abstract

Whale optimization algorithm (WOA) is a recent nature-inspired metaheuristic that mimics the cooperative life of humpback whales and their spiral-shaped hunting mechanism. In this research, it is first argued that the exploitation tendency of WOA is limited and can be considered as one of the main drawbacks of this algorithm. In order to mitigate the problems of immature convergence and stagnation problems, the exploitative and exploratory capabilities of modified WOA in conjunction with a learning mechanism are improved. In this regard, the proposed WOA with associative learning approaches is combined with a recent variant of hill climbing local search to further enhance the exploitation process. The improved algorithm is then employed to tackle a wide range of numerical optimization problems. The results are compared with different well-known and novel techniques on multi-dimensional classic problems and new CEC 2017 test suite. The extensive experiments and statistical tests show the superiority of the proposed BMWOA compared to WOA and several well-established algorithms.

Keywords Nature-inspired computing · Metaheuristic · Optimization · Swarm intelligence

1 Introduction

In the last years, a large number of nature-inspired algorithms have been developed and applied to various kinds of optimization problems [3]. Among such algorithms, swarm-based techniques have been popular, in which the main inspiration is based on a collection of the decentralized and self-organized system of cooperating search agents that have communications with themselves and environment for foraging, direction finding, survival, searching, or hunting [25]. The swarm can be made from the population of animals such as wolves, fishes, birds, salps, ants, and whales [1]. These generic, problem-independent stochastic optimizers can not only tackle problems with a large number of decision variables but also find very competitive and near-optimal results compared to exact traditional exact algorithms [20, 35]. Some of the eminent swarm-based optimizer studied in numerous works are genetic algorithm (GA) [19], differential evolution (DE) [50], and particle swarm optimizer (PSO) [29, 30]. Success of these approaches in handling challenging computational tasks encouraged other researchers for developing many new metaheuristics such as salp swarm algorithm (SSA)

✉ Ali Asghar Heidari
as_heidari@ut.ac.ir

Ibrahim Aljarah
i.aljarah@ju.edu.jo

Hossam Faris
hossam.faris@ju.edu.jo

Huiling Chen
chenhuiling.jlu@gmail.com

Jie Luo
jieluo.cn@gmail.com

Seyedali Mirjalili
seyedali.mirjalili@griffithuni.edu.au

1 School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran

2 Business Information Technology Department, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

3 Department of Computer Science, Wenzhou University, Wenzhou 325035, China

4 Institute of Integrated and Intelligent Systems, Griffith University, Nathan, Brisbane, QLD 4111, Australia

Published online: 29 January 2019

[11, 20], grasshopper optimization algorithm (GOA) [26, 36], and grey wolf optimizer (GWO) [40].

Whale optimization algorithm (WOA) can be regarded as a new efficient metaheuristic inspired by the searching and hunting activities of humpback whales in oceans. This method can outperform well-regarded optimizers such as the PSO, GSA, and GA algorithms [39]. This algorithm shows an efficient performance in dealing with many numerical and real-world optimization tasks. However, similar to other swarm-based metaheuristics [31], this algorithm has been equipped with a number of randomized operators for exploring and exploiting the search space. As such, it may rapidly converge toward a local optimum instead of global best solution when solving optimization problems. This behavior is known as immature convergence and is a widespread problem in metaheuristics [5].

Another common problem is the stagnation in local solutions [27, 28]. This problem can dramatically decrease the quality of the final solutions. These problems may occur when the optimizer cannot make a fine balance between its exploratory and exploitative inclinations [34]. The exploration (global search) can be described as the capability of optimizers in covering and discovering the unknown/untouched regions of the search space on a global scale. The exploitation (local search) phase has to occur after the exploration phase. In this phase, the neighborhood of the better-explored positions in different areas of the problem landscape is covered again to search on a local scale. In fact, after several iterations and checking the global picture of the problem landscape, the optimizer attempts to focus on the sufficiently good explored positions to either increase its chance in finding the optimal solutions or enrich the optimality of the final results. An extensive exploration during the early iterations followed by a focused exploitation in last steps of optimization can assist the optimizer to avoid local solutions [13] A smoother transition from exploration to exploitation can also be considered as a vital factor to increase the overall efficiency of the metaheuristics [24].

It has been observed that some metaheuristics have a better local search (exploitation/intensification) potential [33], some of them show a higher global search (exploration/diversification) tendency [39], and some optimizers have benefits from a good balance between the exploration and exploitation capabilities [22]. Therefore, hybridization of optimizers can alleviate the intrinsic drawbacks of basic algorithms [12]. By this strategy, the hybridized optimizer may inherit all unique advantages of the base algorithms [16]. Reduced calculations, improvement of the precision of results, the stability of optimizer and convergence trend of the basic metaheuristics are some of the main motivations to hybridize optimizers [16].

Up to 2018, the conventional WOA has been applied to several practical applications, and specialized works have reported improved results and performances. In this regard, the basic WOA has been applied to multilevel thresholding image segmentation and the results were satisfactory [17]. The original WOA has been utilized to tackle the optimal renewable resources placement problem in distribution networks [45]. To predict the drug toxicity, Tharwat et al. [48] utilized the basic WOA to attain the best parameters of SVM classifier. The conventional version of WOA has also been employed to find the optimal weights in neural networks [10]. The results of these studies show that the WOA algorithm outperforms several well-regarded optimizers. There are few works in the literature that researchers tried to modify the basic WOA. For instance, WOA has been combined with the simulated annealing (SA) technique for tackling feature selection problems [37].

Oliva et al. [42] utilized a chaos-embedded WOA to deal with parameter estimation of photovoltaic cells, in which chaos were used to enhance the balance between exploration and exploitation powers of WOA and the results confirm the significant impact of the chaos on the efficacy of the WOA. Abdel-Basset et al. [7] proposed a hybrid WOA with local search to deal with flow shop scheduling scenarios. The same authors [6] proposed a levy flight-based WOA to solve bin packing problems through two trials on benchmark cases with different sizes and the results show the improved performance of WOA due to the significant roles of the levy flight. Abdel-Basset et al. [4] also proposed an enhanced discrete WOA with mutation operator and sigmoid and penalty functions to the handle the infeasible solutions. They used the improved WOA for cryptanalysis of Merkle–Hellman knapsack cryptosystem. Another levy flight-based WOA was proposed in [2] to handle 25 different cases of the virtual machine placement problems and the results show that the new variant of WOA can outperform several methods including the WOA, PSO, and GA techniques. Abdel-Basset et al. [8] also proposed an integrated WOA with Tabu search, which is called WAITS, to improve the accuracy and convergence rates of the WOA. The developed method validated using a set of problems on locating hospital departments, and the results confirm the superiority of WAITS variant with acceptable computational time.

The previous works have declared that the WOA can disclose a competent performance in dealing with different problems. The WOA has revealed a relative superior efficacy compared to other well-established optimizers, but it may still be trapped in local solutions. To speed up the convergence leanings, avoid local solutions, and leverage a good balance between the exploratory and exploitative maneuvers, the WOA is hybridized with a new exploratory

β-hill climbing (BHC) local search and an adaptive learning technique in this work.

This paper is organized as follows: the structure of WOA is explained in Sect. 2. Section 3 outlines the BHC technique. The new WOA-based approach is proposed in Sect. 4. Section 5 provides the extensive results and discussions. Finally, conclusions are drawn in Sect. 6.

2 The conventional WOA algorithm

WOA has demonstrated an efficient performance when solving challenging and real-world optimization problems as compared to many well-regarded metaheuristics such as PSO and GSA [37]. The WOA optimizer mimics the idealized hunting activities of humpback whales [39]. These deep-sea hunters search for the victim's position and attack them by encircling the preys or constructing bubble-nets. The exploration phase of WOA is when the whales follow other whales (random leaders) and globally search the prey over a search area. The exploitative phase of WOA is when the whales move directly in direction of the leader to hunt the prey. In WOA algorithm, the best solution in each loop is chosen as the prey (leader).

In the basic WOA, the bubble-net strategy is modeled using spiral motions [42]. The current position of the population is updated according to the helix-shaped movement of the search agents (whales) nearby the target position (prey) as:

$$x(t+1) = D.e^{bl}\cos(2\pi l) + x_p \tag{1}$$

where $D = |x_p(t) - x(t)|$ shows the distance among a whale ($x(t)$) and the prey ($x_p(t)$), l is a random number inside [− 1,1], b shows a constant, t denotes the current iteration, and (.) shows the multiplication of elements. Alternatively, during the encircling policy, the search agents update their locations with regard to the best position as:

$$D = |C.x_p(t) - x(t)| \tag{2}$$

$$x(t+1) = x_p(t) - A.D \tag{3}$$

where D denotes the distance among the whales and victim. The C and A denote the coefficient vectors, which are calculated as:

$$C = 2r \tag{4}$$

$$A = 2ar - a \tag{5}$$

where r shows a random value, and a is linearly dropped from 2 to 0 as the iterations continue. The hunting strategy of whales in 2D and 3D space is demonstrated in Figs. 1 and 2, respectively.

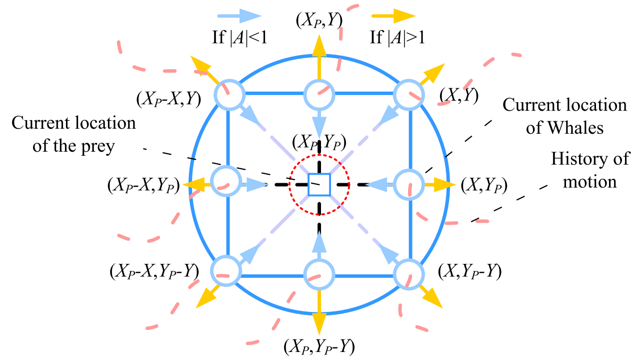

Fig. 1 2D view of shrinking encircling strategy of whales

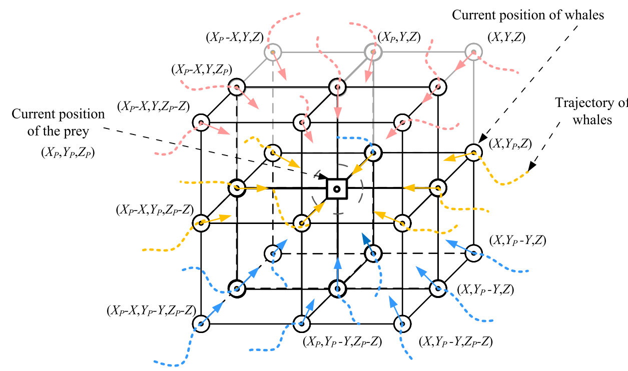

Fig. 2 3D view of hunting strategy

The search agents can simultaneously spin nearby their quarry based on a shrinking ring and along a spiral formed trajectory. This procedure is performed based on:

$$x(t+1) = \begin{cases} x_p(t) - A.D, & p < 0.5 \\ D.e^{bl}\cos(2\pi.l) + x_p(t), & p \geq 0.5 \end{cases} \tag{6}$$

where p is inside (0, 1) and shows the chance of each strategy. In some situations, real whales search for the prey randomly. To model this fast in WOA, the position vector of whales is occasionally updated with regard to a random whale instead of the top search agent as:

$$D = |C.x_{rand}(t) - x(t)| \tag{7}$$

$$x(t+1) = x_{rand}(t) - A.D \tag{8}$$

where $x_{rand}(t)$ is a randomly selected location in the present population.

Algorithm 1 shows the pseudo-code of the original WOA algorithm.

Algorithm 1 Pseudo-code of WOA

```
Set the swarm size N and maximum iterations (T)
Initialize a, A, C and the random position of whales X_i
Compute the fitness of search agents
Set x_p as the best solution (leader)
While (t < T) do
    Update a, A, C, l and p
    for each whale
        if p < 0.5 then
            if |A| > 1 then
            Determine a random leader x_rand
            Update the locations using Eq. (8)
            elseif |A| < 1 then
            Update the locations using Eq. (3)
            end if
        elseif p > 0.5 then
        Update the locations using Eq.(1)
        end if
    end for
    Check the space limits
    Update the leader x_p if there is a new better solution
    t=t+1
end while
return the leader x_p
```

3 The β-hill climbing (BHC) algorithm

The hill climbing (HC) technique can be considered as one of the most conventional local search (exploitation) strategies [9]. The HC can iteratively enrich a series of randomly approximated solutions by moving them toward better adjacent localities [23]. The search continues until it reaches a relatively best locally optimal solution. The HC only keeps the downhill solutions with improved qualities compared to the uphill points. For this reason, it has a fast convergence speed in exploiting local solutions. However, it is inclined to stagnation to local solutions and for this reason, it cannot find the globally best optimum for challenging multimodal problems [9]. As a result, researchers revised HC by new stochastic operators to improve its performance regarding LO stagnation drawbacks [49].

The basic HC cannot perform an exploratory search and consequently avoid local solutions in the case of immature convergence. In 2017, Azmi Al-Betar proposed an extended HC with exploratory behaviors, which is called β-hill climbing technique (BHC) [9]. In the BHC, a stochastic strategy entitled β-operator is employed to establish a fine balance among the exploration and exploitation throughout the global search. The origin of the idea for the new operator in BHC is inspired from the uniform mutation in the famous GA technique [9].

The BHC begins the exploitation with a random point $x = (x_1, x_2, x_3, \ldots, x_N)$. It utilizes N-operator (neighborhood navigation) and β-operator throughout its exploitative steps to create a fresh solution $x' = (x'_1, x'_2, x'_3, \ldots, x'_N)$. In the N-operator, a randomly selected adjacent point of x should be adopted as:

$$x'_i = x_i \pm U(0,1) \times bw, i \in [1, 2, 3, \ldots, N] \tag{9}$$

where the factor bw represents the bandwidth between the existing point and the fresh one. It's worth noting that i is a randomly chosen index from the dimensionality interval [1, N]. In β-operator phase, the elements of the new positions are updated according to its current values or filled randomly based on the searching domain with a chance of β as:

$$x'_i \leftarrow \begin{cases} x_r & z < \beta \\ x_i & \text{otherwise} \end{cases} \tag{10}$$

where x_i shows the decision variable and z denotes a random number inside (0, 1). Then, the BHC compares the x'_i to x_i and only records the superior (downhill) one:

$$x_i \leftarrow \begin{cases} x'_i & f(x'_i) \leqslant f(x_i) \\ x_i & \text{otherwise} \end{cases} \tag{11}$$

The exploration and exploitation phases of the BHC are controlled by its operators. The N-operator can navigate the neighboring points of the current solution. Then, it randomly selects the better solution from the downhill. The β-operator assists BHC in converging to better-quality points by constructing a series of the elements of the existing position, and as a result, the convergence trend of BHC can possibly be accelerated. The β-operator is responsible for the exploration phase of the BHC while the N-operator can perform more exploitative steps in the neighborhood of the superior solutions. In terms of exploration, the BHC is capable of 'jumping out' from local solutions using the β-operator that can be seen as the engine of avoiding local solutions. Therefore, the BHC can be utilized as an exploratory local search.

Algorithm 2 demonstrates the pseudo-code of the BHC local search.

Algorithm 2 Pseudo-code of the BHC

```
Set the dimension N, initial probabilities β and bw and maximum iterations (T)
Initialize the initial solution x_i=lb_i+(ub_i − lb_i) × U(0,1)
Compute the f(x)
While (t < T) do
    perform Eq.(9)
        for i =1,2,...,N do
            set a random value inside (0, 1) to z
            if z ⩽ β then
            x'_i=lb_i+(ub_i − lb_i) × U(0,1)
            end if
        end for
    if f(x') ⩽ f(x) then
    x=x'
    end if
    t=t+1
end while
return the best solution
```

4 The proposed BMWOA

In the literature, the satisfactory efficacy of the WOA has been verified in tackling several practical tasks [17]. The WOA technique has an acceptable exploration (diversification) capacity because of the role of random leaders. These random front-runners can deepen the diversification of crowd throughout the early stages of the hunt. However, the basic WOA can still be improved in terms of either exploration or exploitation tendencies. The background idea here is to preserve the unique simplicity of the WOA and also enrich its exploratory and exploitative features.

In this section, a new beta version of the WOA, which is called BMWOA, is developed to alleviate the drawbacks identified above. In fact, two new effective mechanisms are considered to reduce the probability of immature convergence and local optima stagnation of the original WOA. First, the BMWOA optimizer utilizes a simple operator to model associative learning and instantaneous memory of whales all through the exploration stage. Then, it is hybridized with the BHC local search. The description of the new operators is provided in the next subsections.

4.1 WOA with associative learning and memory

Humpback whales accomplish team foraging strategies to hunt the victim, yet the details about exactly how individuals can harmonize their behaviors with other predators in the hunting team are challenging to discover [43]. Recent investigations in the social life of these marine mammals revealed that the whales can learn from other members of the group and use acoustic communication techniques [18]. Although, cooperative organization of whales differs from single hunters to steady short-term associations among both relatives and non-kin [44], behavioral investigations of whales have exposed fast social broadcast of acoustic repertoire and social learning backing the spread of new chasing activities inside a group [46].

From an optimization perspective, different types of learning techniques have also been utilized in previous optimizers to pass on useful info from one generation to another one. Inspired by the aforementioned facts and the learning operators of other metaheuristics such as moth swarm (MSA) [41] and PSO [14] algorithms, a new learning tactic is embedded in the proposed WOA during learning condition when $p < 0.5$, $|A| > 1$, and $q > 0.5$, where q is another random inside (0, 1). To perform the learning phase, each whale should randomly learn from either other whales or the leader. By this rule, whales can approach the victim by learning from other whales and the leader at the same time. The STM also assists whales in following the prey by using their recently learned hunting behaviors. To simulate the STM, it is generated in each step based on a uniform Gaussian distribution on an interval from each up-to-date whale to the boundaries of the search space. Therefore, the new location of whales during the learning step is attained using Eq. (12):

$$x_i(t+1) = x_i(t) + 0.001G(x_i(t) - lb, ub - x_i(t)) + S_1 r_1 (x_r(t) - x_i(t)) + S_2 r_2 (x_p(t) - x_i(t)) \tag{12}$$

where r_1 and r_2 are random values in (0, 1), x_r denotes a random leader from the preceding generation, S_1 and S_2 represent adaptive cognitive and social factors, respectively. The dynamics S_1 and S_2 are updated in each iteration using:

$$S_1 = (1 - t/T) \tag{13}$$

$$S_2 = 2(t/T) \tag{14}$$

As iteration increases, the value of adaptive cognitive parameter S_1 raises to reduce the impact of the random leader on the search patten of whales. On the other hand, the value of S_2 increases to enhance the impact of the leader whale. The reason is that the rule is designed to approach the victim by learning from other whales and the leader simultaneously. Therefore in the last steps, the agents need to uninterruptedly approach to the high-quality solution (victim) rather than the position of other agents. In this manner, during the course of iterations, the median of the population will converge to better-quality positions.

The basic WOA attempts to explore diverse zones of the space when $p < 0.5$, $|A| > 1$. In this condition, whales make steps toward randomly selected agents. In BMWOA, when $p < 0.5$, $|A| > 1$, and random $q > 0.5$, the location vector of whales is updated similar to WOA using Eq. (8). As an alternative tactic during the global search, when $q > 0.5$, the positions will be reorganized based on the earlier random leader, best leader, and a random Gaussian term devoted to the STM of each whale. This policy utilizes random q and a learning structure. Therefore, it can further put emphasis on random behaviors of the proposed BMWOA. It encourages far-reaching exploration of the unknown search realm. Additionally, the BMWOA can randomly switch between two exploration proclivities and it enriches the overall searching (hunting) patterns of BMWOA because it is well-informed about the gen delivered by each whale and the top leader (global best). Note that the BMWOA will inherit the exploratory traits of the WOA when $q < 0.5$.

4.2 The BMWOA with BHC local search

To improve the exploitation proclivity and convergence trend of the modified BMWOA, the BHC local search is also embedded into the algorithm when yet $p < 0.5$, but $|A| < 1$ and $r > 0.5$, where r is an added random inside (0, 1).

For this purpose, the vicinity of the latest solution vector is deeply exploited using the core operators of BHC. As explained in Sect. 3, the BHC will execute two consecutive operators on each whale to explore and exploit new regions around it. Then, it only keeps the superior downhill one from two competing agents. The winner whale will be further improved during the next stages while the other one is eliminated. In the N-operator, the locations of whales are updated based on:

$$x_i' = x_i + bw \times (2r - 1), \quad i \in [1, 2, 3, \ldots, N] \tag{15}$$

where r is a random value inside (0, 1). Then, the β-operator should be applied to all whales based on:

$$x_i' \leftarrow \begin{cases} x_r & z < \beta \\ x_i & \text{otherwise} \end{cases} \tag{16}$$

The better whale will be selected and inserted to the next stages based on the S-operator:

$$x_i \leftarrow \begin{cases} x_i' & f(x_i') \leqslant f(x_i) \\ x_i & \text{otherwise} \end{cases} \tag{17}$$

In fact, the new operators act like an exploratory local search machine that first receives a whale. Then, the engine, as an exploitation tool, attempts to inform the whale from its neighborhood areas. When the condition of β-operator is met, it considers an additional virtual random whale over the search domain. Then, it compares the locally informed whale to its random pair based on the fitness value and records the fitter one (downhill whale) as the new improved solution.

Note that when $p < 0.5$, $|A| < 1$ and $r < 0.5$, the whales move toward the leader similar to the WOA using Eq. (2). When $p > 0.5$, the BMWOA also updates the whales similar to the WOA based on Eq. (1).

Algorithm 3 shows the pseudo-code of the proposed BMWOA optimizer.

The below remarks may help to understand why the new modifications can be potentially beneficial:

- In the case of early stagnation to locally optimal solutions, the BMWOA can escape from them because it has an improved exploration engine with enriched searching patterns. The whales have more chance to be distributed over the solution space regarding both the adaptive learning tactic and the role of random leaders.
- The BMWOA can show the exploratory and exploitative advantages of the WOA because it uses several randomized switching factors to also include all operators of the original WOA.
- The BMWOA can establish a more stable trade-off between exploration and exploitation trends in dealing with target landscapes because the new operators assist BMWOA to smoothly shift from the improved global search in first steps to exploitation in later phases.
- The BMWOA has a fast convergence because it also has an improved BHC-based local search when it finds a promising solution.
- The S-operator only retains improved solutions during the BHC-based exploitation phase; hence, the quality of the found results and speed of convergence can be further improved in the BMWOA.
- In the case of entrapment in LO in any stage of the searching, the BMWOA has an improved potential to escape from them because it also utilizes an exploratory local β-operator in last steps.

Algorithm 3 Pseudo-code of the BMWOA

```
Set the initial parameter β and bw, swarm size N and maximum iterations (T)
Initialize a, A, C and the random position of whales X_i
Compute the fitness of search agents
Set x_p as the best solution (leader)
While (t < T) do
    Update a, A, C, l and random numbers p,q,r
    for each whale do
        if p < 0.5 then
            if |A| > 1 then
                if q < 0.5 then
                    Determine a random leader x_rand
                    Update the locations using Eq. (8)
                elseif q > 0.5 then
                    Update the locations using Eq. (12)
                end if
            elseif |A| < 1 then
                if r < 0.5 then
                    Update the locations using Eq. (3)
                elseif r > 0.5 then
                    Perform the N-operator using Eq. (15)
                    Perform the β-operator using Eq. (16)
                    Perform the S-operator using Eq. (17)
                end if
        elseif p > 0.5 then
            Update the locations using Eq.(1)
        end if
    end for
    Check the space limits
    Update the leader x_p if there is a new better solution
    t=t+1
end while
return the leader x_p
```

4.3 Computational complexity

Note that the computational complexity of BMWOA depends on the number of whales (n), the dimensions (d), and the maximum number of iterations (t). Therefore, the overall computational complexity is O(BMWOA) = O(Initialization) $+\ t \times$(O(Fitness evaluation of all whales) + O(Updating process of all whales with new mechanisms)). The computational complexity of evaluating the fitness of all agents is O($n \times d$). Updating the position of all agents with new mechanism is O($(\frac{1}{4}) \times n \times d^2$). Therefore, the overall computational complexity of the BMWOA is of O($n \times d + (\frac{1}{4}) \times n \times d^2$).

5 Experimental results and discussions

In this section, the efficiency of the proposed BMWOA optimizer is verified and compared to the basic WOA and several other well-established and new optimizers using

Table 1 Description of unimodal benchmark functions

Function	Dimensions	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} x_i^2$	10, 30, 50, 100	[− 100, 100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	10, 30, 50, 100	[− 10, 10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	10, 30, 50, 100	[− 100, 100]	0
$f_4(x) = \max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	10, 30, 50, 100	[− 100, 100]	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	10, 30, 50, 100	[− 30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	10, 30, 50, 100	[− 100, 100]	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	10, 30, 50, 100	[− 128, 128]	0

Table 2 Description of multimodal benchmark functions

Function	Dimensions	Range	$f_{\min}$
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	10, 30, 50, 100	$[-500, 500]$	$-418.9829 \times n$
$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	10, 30, 50, 100	[− 5.12, 5.12]	0
$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	10, 30, 50, 100	[− 32, 32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	10, 30, 50, 100	[− 600, 600]	0
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\}$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)\ y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	10, 30, 50, 100	[− 50,50]	0
$f_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right]\right.$ $\left.+(x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	10, 30, 50, 100	[− 50, 50]	0

Table 3 Description of fixed-dimension multimodal benchmark functions

Function	Dimensions	Range	$f_{\min}$
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}\left(x_i - a_{ij}\right)^6}\right)^{-1}$	2	$[-65, 65]$	1
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]$	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	− 1.0316
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398
$f_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$ $\times\left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	$[-2, 2]$	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	$[1, 3]$	− 3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	$[0, 1]$	− 3.32
$f_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	− 10.1532
$f_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	− 10.4028
$f_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	− 10.5363

Table 4 Sensitivity of the proposed BMWOA to the values of β and *bw* parameters (best results are shown in bold)

Case		P1	P2	P3	P4	P5	P6	P7	P8	P9
Beta		0.5	0.5	0.5	0.05	0.05	0.05	0.005	0.005	0.005
bw		0.5	0.05	0.005	0.5	0.05	0.005	0.5	0.05	0.005
F1	Avg	0.000298	0.000434	0.000337	1.32E−06	7E−06	4.07E−06	**9E − 23**	3.26E−21	2.36E−20
	Stdv	0.000479	0.00069	0.00066	3.76E−06	1.98E−05	5.06E−06	1.47E−22	1.1E−20	5.4E−20
F2	Avg	0.008621	0.010551	0.00925	5.68E−09	3.61E−07	4.37E−07	**9.18E − 17**	1.57E−16	2.79E−15
	Stdv	0.009052	0.010001	0.008908	2.98E−08	1.54E−06	1.07E−06	1.25E−16	2.56E−16	2.72E−15
F3	Avg	0.065745	0.067219	0.108953	0.06739	**0.061433**	0.127947	1.954173	0.991972	1.296849
	Stdv	0.086301	0.101484	0.160351	0.16159	0.134719	0.333195	5.612603	2.450952	3.258599
F4	Avg	0.003017	0.002635	**0.002372**	0.007093	0.0053	0.004886	0.03995	0.058537	0.044716
	Stdv	0.002612	0.002275	0.002743	0.008074	0.005453	0.005344	0.045594	0.057331	0.049104
F5	Avg	0.004285	**0.003612**	0.00381	0.009372	0.007749	0.014398	10.81694	9.154318	2.49777
	Stdv	0.009132	0.006117	0.006932	0.017785	0.012131	0.042051	12.55148	12.20859	7.564064
F6	Avg	0.000868	0.000412	0.000267	0.000413	0.000444	0.000921	**1.7E − 05**	2.5E−05	2.27E−05
	Stdv	0.001456	0.000891	0.000536	0.000579	0.000598	0.001139	9.5E−06	1.62E−05	1.25E−05
F7	Avg	0.001451	0.001138	0.001459	0.001349	0.000918	0.001219	0.00111	**0.000857**	0.000977
	Stdv	0.001408	0.000864	0.001174	0.000923	0.0009	0.000952	0.001144	0.000711	0.000992
F8	Avg	**− 12,569.5**	**− 12,569.5**	**− 12,569.5**	**− 12,569.5**	**− 12,569.5**	**− 12,569.5**	**− 12,569.4**	**− 12,569.4**	**− 12,569.4**
	Stdv	0.00298	0.001368	0.002926	0.005021	0.022539	0.015036	0.098531	0.156792	0.109605
F9	Avg	0.00029	0.000199	0.000296	2.88E−07	1.55E−06	1.56E−05	1.31E−13	0.000124	**3.22E–14**
	Stdv	0.000546	0.000258	0.000584	4.34E−07	2.39E−06	3.45E−05	2.85E−13	0.000679	4.65E−14
F10	Avg	0.004859	0.004084	0.003057	2.01E−05	0.000242	0.000629	**7.02E − 13**	2.49E−12	1.3E−11
	Stdv	0.005512	0.003614	0.004532	3.59E−05	0.000408	0.00063	1.84E−12	3.74E−12	1.26E−11
F11	Avg	0.000762	0.001156	0.002368	**6.54E − 06**	3.3E−05	4.74E−05	0.003006	0.003888	0.006042
	Stdv	0.000883	0.002846	0.006366	1.36E−05	6.61E−05	7.74E−05	0.006896	0.011297	0.013345
F12	Avg	3.31E−06	3.8E−06	3.45E−06	1.41E−05	1.3E−05	1.06E−05	3.12E−06	3.61E−06	**3.12E − 06**
	Stdv	6.29E−06	7.06E−06	5.57E−06	2.36E−05	2.98E−05	1.72E−05	2.06E−06	2.56E−06	2.67E−06
F13	Avg	5.73E−05	**2.49E − 05**	4.83E−05	9.1E−05	8.27E−05	0.000135	0.001538	0.001142	0.001161
	Stdv	0.000159	3.91E−05	7E−05	0.000149	0.000114	0.000459	0.003877	0.003364	0.003388
F14	Avg	**0.998004**	**0.998004**	**0.998004**	**0.998004**	**0.998004**	**0.998004**	**1.196414**	**1.130277**	**1.262551**
	Stdv	7.62E−12	1.8E−11	2.41E−11	5.59E−16	6.37E−16	5.14E−16	0.605406	0.503383	0.685995
F15	Avg	0.000416	0.00039	0.000371	0.000403	**0.000339**	0.00043	0.000503	0.000494	0.000495
	Stdv	0.000171	0.000118	5.27E−05	0.000293	0.000167	0.000317	0.000403	0.000371	0.000381
F16	Avg	−1.0316	−1.03161	−1.0316	**− 1.03163**	**− 1.03163**	**− 1.03163**	**− 1.03163**	**− 1.03163**	**− 1.03163**
	Stdv	3.11E−05	2.3E−05	5.45E−05	1.11E−14	2.58E−14	9.1E−15	4.99E−14	7.84E−14	9.31E−14
F17	Avg	0.397908	0.397921	0.397899	**0.397887**	**0.397887**	**0.397887**	**0.397887**	**0.397887**	**0.397887**
	Stdv	2.5E−05	4.19E−05	1.57E−05	6.54E−12	1.47E−12	1.8E−12	1.77E−12	1.94E−12	1.39E−12
F18	Avg	3.000077	3.00008	3.000171	**3**	**3**	**3**	**3**	**3**	**3**
	Stdv	9.59E−05	9.34E−05	0.000299	3.05E−12	4.72E−12	3.17E−11	8.77E−12	9.01E−12	1.18E−11
F19	Avg	−3.86227	−3.8622	−3.86184	**− 3.86278**	**− 3.86278**	**− 3.86278**	**− 3.86278**	**− 3.86278**	**− 3.86278**
	Stdv	0.000698	0.000971	0.001266	1.33E−10	3.89E−10	4.87E−10	1.56E−10	1.03E−10	8.27E−11
F20	Avg	−3.21717	−3.21194	−3.22403	−3.25859	−3.26651	**− 3.28633**	−3.25444	−3.27444	−3.25817
	Stdv	0.0727	0.086642	0.072562	0.060329	0.060329	0.055415	0.060098	0.059241	0.060739
F21	Avg	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**
	Stdv	4.66E−05	1.9E−05	8.27E−06	8.82E−08	1.24E−07	1.91E−07	1.61E−05	2.27E−05	3.56E−08
F22	Avg	−10.4028	−10.4028	−10.4028	**− 10.4029**	**− 10.4029**	**− 10.4029**	**− 10.4029**	**− 10.4029**	**− 10.4029**
	Stdv	2.51E−05	3.36E−05	1.03E−05	1.17E−07	2.42E−07	1.53E−07	2.3E−08	4.38E−08	6.62E−08
F23	Avg	−10.5363	−10.5363	−10.5363	**− 10.5364**	**− 10.5364**	**− 10.5364**	−10.1759	**− 10.5364**	**− 10.5364**
	Stdv	1.26E−05	1.54E−05	1.57E−05	1.22E−07	4.96E−08	1.94E−07	1.372036	8.86E−08	2.92E−08

various benchmark problems. All utilized codes in this study have been implemented in the same manner using MATLAB 7.10 (R2010a) and run on a PC with the Windows 7 64-bit professional and 64 GB RAM. These extensive evaluative tests are divided into several experimental phases; first, a sensitivity analysis is carefully accomplished to assess the effect of β and bw factors on the results of the BMWOA. To investigate the influence of the dimensions on the efficacy of the WOA and BMWOA, a comparative scalability assessment is also accomplished. Then, the efficiency of BMWOA in dealing with different benchmark sets is deeply compared to other metaheuristics.

5.1 The benchmark set 1

The proposed BMWOA approach is substantiated based on 23 classical benchmark tasks listed in Tables 1, 2, and 3. This well-regarded benchmark suite contains totally 23 unimodal and multimodal problems with various dimensions and landscapes with dissimilar difficulty levels including seven scalable unimodal, six scalable multimodal, and 10 fixed-dimension multimodal functions. This makes them suitable for benchmarking the exploratory and exploitative capacities of optimizers.

5.2 Sensitivity analysis

The proposed BMWOA has two user-defined parameters related to the utilized BHC local search in the exploitation stage. Here, the best settings for the internal parameters of BHC local search is determined using a series of comparative studies on F1–F23 problem with different choices for β and bw values. Nine different sets for the possible options for selecting the β and bw parameters are considered as P1–P9 settings. The effect of P1–P9 settings on the quality of search is investigated by changing the parameters and recording the average (Avg) and standard deviation (Stdv) of the estimated results. Table 4 shows the results obtained with different settings. From Table 4, it can be recognized that the best setting is P7. Hence, in the rest of the tests, the β and bw parameters are set to 0.005 and 0.5, respectively.

5.3 Comparative study

In this part, the efficacy of the BMWOA in terms of standard deviation (Stdv) and average (Avg) results is compared to several other well-regarded metaheuristics such as the GA [47], PBIL [47], FPA [52], PSO [47], BBO [47], GWO [40], BAT [51] and FA [21] algorithms. In addition, it is compared to the basic WOA [39], and the recent SCA [38]. Parameters of GA, PSO, PBIL, and BBO are set based on the original work of BBO algorithm in [47], while the rest of algorithms are set based on the recommended settings in their original papers. Details of parameters are shown in Table 5. For these experiments, the number of function evaluation is set to 25E+03 and dimension of F1–F13 problems is set to 30. The presented results are recorded based on 30 independent runs with random initial conditions.

In order to investigate significant differences of obtained results for the BMWOA over other competitors, nonparametric Wilcoxon rank-sum test [15] at 5% significance level was also employed in this paper. The p values of comparisons are reported in Table 9. In Table 9, the best algorithm for each case is tested against other optimizers.

Table 6 reports the results of the developed BMWOA versus other evaluated algorithms for F1–F7 problems. The average convergence inclinations (average error results) of

Table 5 The parameter settings

Algorithm	Parameter	Value
GA	Single point crossover	1
	Mutation	0.01
	Roulette wheel selection	
PSO	Topology fully connected	
	Inertia factor	0.3
	c_1	1
	c_2	1
GWO	a	[2 0]
SCA	a	2
WOA	a	[2 0]
	a_2	[− 2 −1]
	b	1
PBIL	Learning rate	0.05
	Good population member	1
	Bad population member	0
	Elitism parameter	1
	Mutational probability	0.1
BAT	$Q_{\min}$ frequency minimum	0
	$Q_{\max}$ frequency maximum	2
	A loudness	0.5
	r pulse rate	0.5
FA	α	0.5
	β	0.2
	γ	1
FPA	Probability switch p	0.8
BBO	Habitat modification probability	1
	Immigration probability limits	[0, 1]
	Step size	1
	Max immigration (I) and Max emigration (E)	1
	Mutation probability	0.005

Table 6 The results for F1–F7 problems

Function	Metric	PBIL	GA	PSO	BBO	FPA	GWO	BAT	FA	SCA	WOA	BMWOA
F1	Avg	47,231.067	1418.05	16,853.61	22.0328	2753.22	2.3E−33	60,127.814	0.00585	2.96253	3.9E−97	4E−210
	Stdv	1901.8083	594.442	931.8746	7.50303	409.889	3.4E−33	7048.9538	0.00255	3.69978	1.6E−84	0
	Rank	10	7	9	6	8	3	11	4	5	2	1
F1F2	Avg	98.9	24.766	102.6122	0	43.3278	6.2E−20	2,114,698.4	0.37677	0.01036	1.3E−60	0
	Stdv	2.687626	5.24442	180.6911	0	8.3243	3.8E−20	7,469,401.9	0.11081	0.01942	4.3E−53	0
	Rank	9	7	10	1	8	4	11	6	5	3	1
F1F3	Avg	55,352.867	22,229.6	35,391.2	9750.19	2929.04	5.2E−08	110,443.17	1552.8	7082.42	8424.27	4E−213
	Stdv	5535.1238	4485.23	5551.04	1398.53	682.717	1.1E−07	44,903.3	596.633	4695.42	11,277.4	0
	Rank	10	8	9	7	4	2	11	3	5	6	1
F1F4	Avg	79.733333	51.3043	47.96845	24.2261	30.6416	2.1E−08	83.159008	0.09684	26.2128	0.2316	4E−102
	Stdv	1.5040686	6.46929	2.020522	3.4669	3.74592	2.2E−08	3.1368021	0.02588	8.57487	25.7729	2E−101
	Rank	10	9	8	5	7	2	11	3	6	4	1
F1F5	Avg	139,706,436	6955.84	14,734,699	1009.13	777,796	26.6971	177,734,246	160.652	12,303.1	26.7274	10.8169
	Stdv	20,603,441	9790.34	4,569,491	419.385	370,803	0.75791	30,470,262	323.809	20,760.5	0.38263	12.3405
	Rank	10	6	9	5	8	2	11	4	7	3	1
F1F6	Avg	38,284.167	959.706	17,244.97	32.1263	2913.53	0.41474	61,906.789	0.00642	7.38408	0.01524	1.7E−05
	Stdv	9466.4907	251.913	1150.569	7.48097	856.936	0.27191	7188.7243	0.00272	4.22115	0.04926	9.3E−06
	Rank	10	7	9	6	8	4	11	2	5	3	1
F1F7	Avg	255.21996	0.15658	5.509702	0.00277	0.64389	0.00119	30.674805	0.07488	0.0729	0.00111	4.5E−05
	Stdv	28.603828	0.07351	2.447328	0.0019	0.2353	0.00068	5.0110319	0.0423	0.14019	0.00112	0.0027
	Rank	11	7	9	4	8	3	10	6	5	2	1
Sum		70	51	63	34	51	20	76	28	38	23	7
Overall		10	7	9	5	7	2	11	4	6	3	1

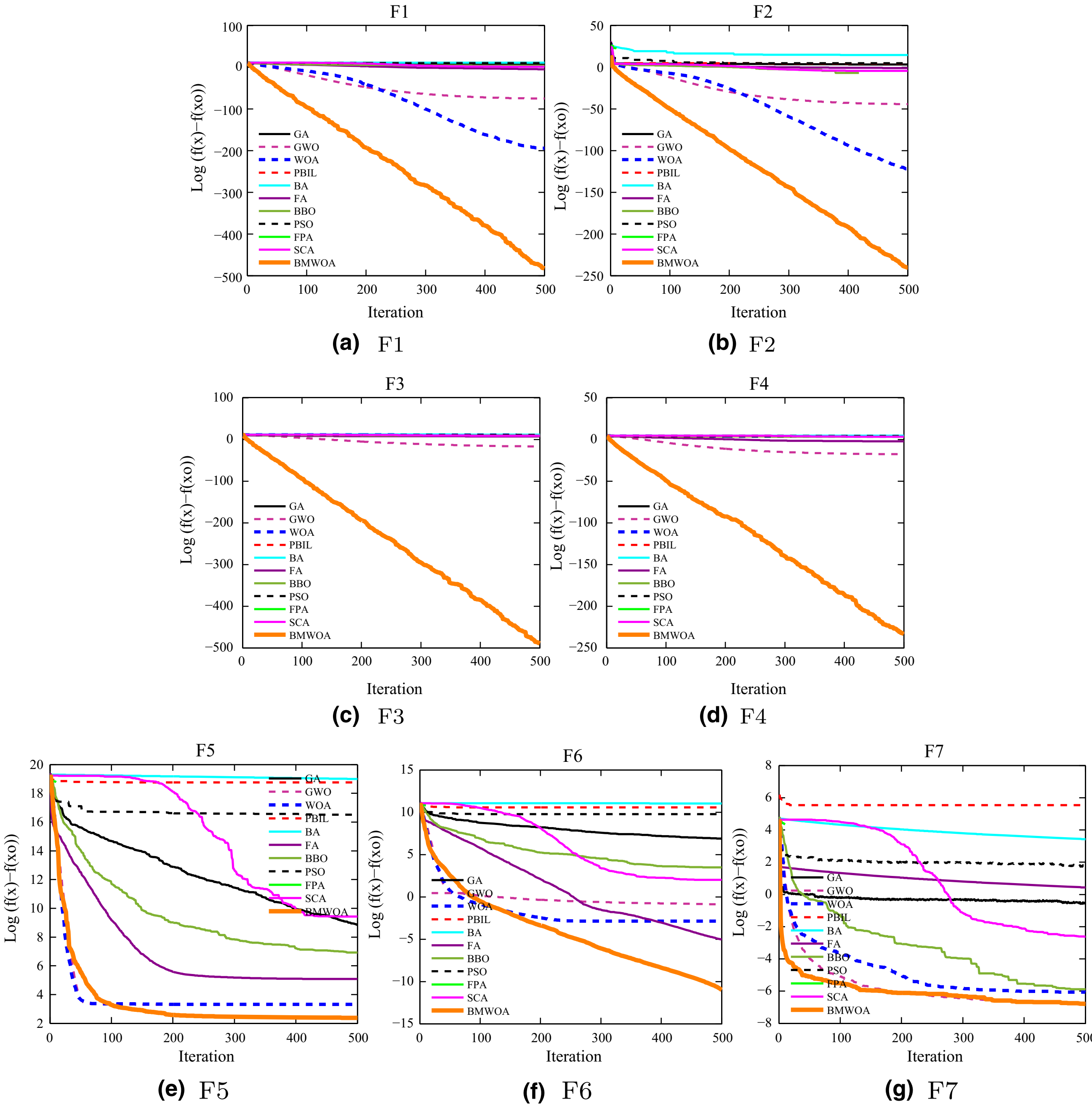

Fig. 3 Convergence curves for (F1–F7) problems

all optimizers in logarithmic scale are also compared in Fig. 3.

From the results in Table 6, it is observed that the proposed BMWOA can be recognized as the best optimizer when solving unimodal F1–F7 problems. It has attained the finest solutions for all F1–F7 problems. According to the concluding ranks, and after BMWOA, the GWO, WOA, FA, BBO, and SCA techniques have reached the next places. The well-known F1–F7 problems can examine the intensification/exploitation propensities of these optimizers. The reason for the superior performance of BMWOA on unimodal F1–F7 is that this enriched technique is equipped with heightened exploitative trends due to the embedded BHC local search in addition to the exploitative patterns inherited from the original WOA. Therefore, once it catches a downhill solution inside the allowable bounds, it can keep it and continue the search in the vicinity of that point for exploiting better-quality solutions, while other algorithms such as BAT, SCA or FPA are not capable of exploiting the search space to this

Table 7 The results for F8–F13 problems

Function	Metric	PBIL	GA	PSO	BBO	FPA	GWO	BAT	FA	SCA	WOA	BMWOA
F8	Avg	− 4345.598	− 12,564	− 4094.45	− 12,517	− 6574.2	− 5847.2	− 2581.937	− 6174.9	− 3833	− 12,569	− 12,569
	Stdv	353.24085	0.9748	181.747	12.6566	202.956	945.374	454.63122	1036.84	224.665	1772.13	0.09688
	Rank	8	3	9	4	5	7	11	6	10	2	1
F9	Avg	141.5	15.7212	294.5026	0	193.406	1.37334	180.72012	41.7507	27.5773	1.3E−13	0
	Stdv	14.023195	5.10838	7.51532	0	14.4361	3.02405	25.286287	12.02	18.8681	2.8E−13	2E−14
	Rank	8	5	11	1	10	4	9	7	6	3	1
F10	Avg	18.393282	14.4338	17.33955	1.21779	9.49562	4.3E−14	18.99322	0.04623	12.1571	7E−13	8.9E−16
	Stdv	0.0480885	0.84536	0.372317	0.21347	1.21405	4.5E−15	0.2270931	0.01851	9.7132	1.8E−12	2.7E−15
	Rank	10	8	9	5	6	2	11	4	7	3	1
F11	Avg	433.38503	15.2496	155.0289	1.42119	29.1752	0.00301	568.084	0.00467	0.73315	0.00301	0
	Stdv	33.231149	7.30361	18.93677	0.19589	6.74777	0.00806	67.473418	0.00273	0.31265	0.00678	0.0369
	Rank	10	7	9	6	8	3	11	4	5	2	1
F12	Avg	248,076,406	3.07885	9530031	0.16374	13,122.7	0.02963	410,480,518	0.00045	12,324.5	0.00088	3.1E−06
	Stdv	22,869,836	1.82823	3,031,759	0.12542	18,675.7	0.01514	81,175,799	0.00052	55,432.5	0.00709	2E−06
	Rank	10	6	9	5	8	4	11	2	7	3	1
F13	Avg	156,925,367	8.70577	17,486,233	0.69157	898,699	0.37171	420,229,131	0.0029	342.456	0.00796	0.00154
	Stdv	16,722,325	1.27777	10,533,176	0.11137	803,448	0.19146	109,962,596	0.0024	1433.27	0.10703	0.00381
	Rank	10	6	9	5	8	4	11	2	7	3	1
Sum		56	35	56	26	45	24	64	25	42	16	6
Overall		9	6	9	5	8	3	11	4	7	2	1

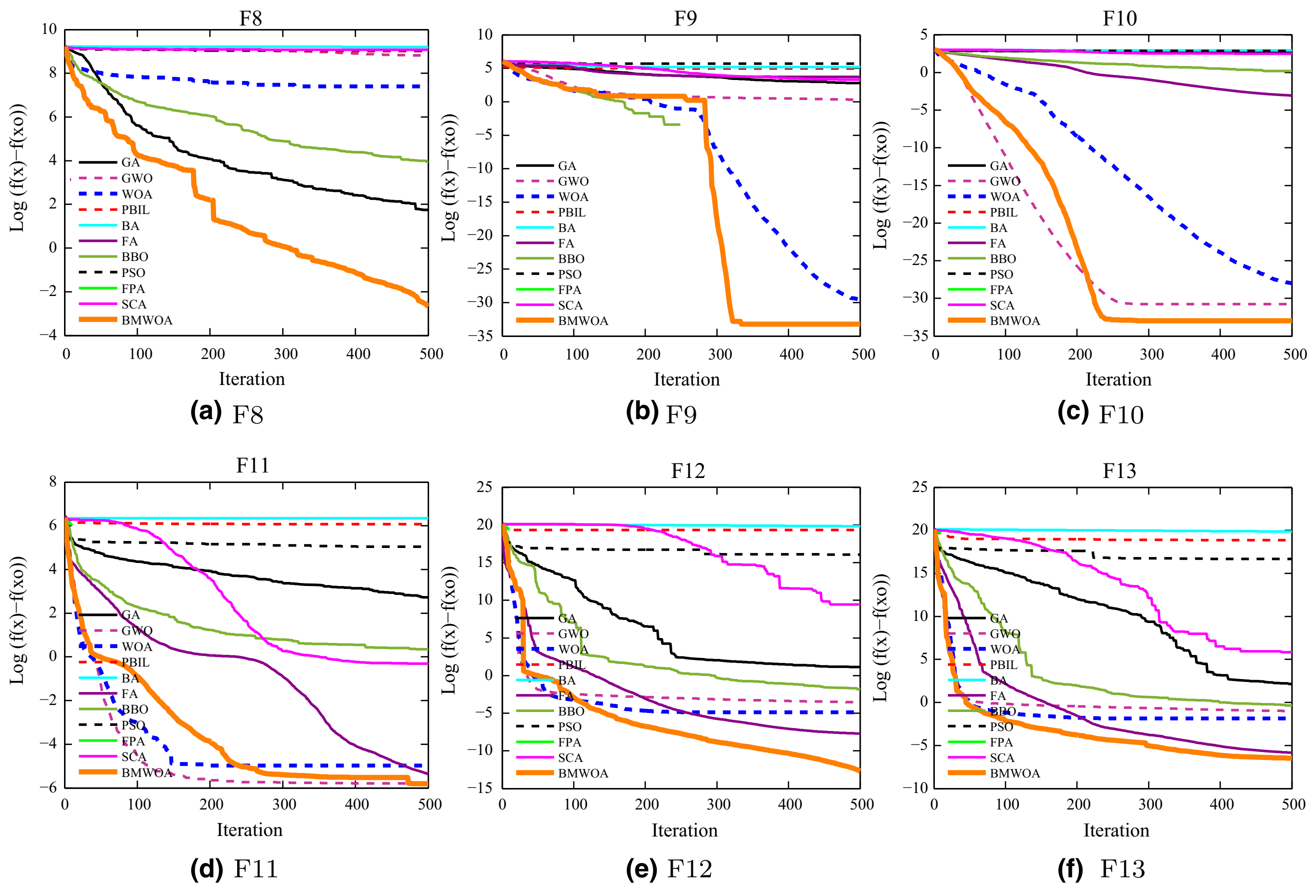

Fig. 4 Convergence curves for (F8–F13) problems

level and based on these deep-rooted exploitative mechanisms. It is seen that WOA and GWO algorithms also demonstrate good performances on F1 and F5. Inspecting the results for F2, it is seen that both BMWOA and BBO outperform other approaches with finding the exact optimum. Inspecting the inclinations in Fig. 3, it is observed that the BMWOA is the most accelerated technique among other methods on F1–F7, whereas some stagnation behaviors can be detected in the curves of other solvers such as PSO, BAT, and FPA.

Table 7 reveals the Avg, Stdv and ranking results of the proposed BMWOA versus other algorithms in dealing with multimodal F8–F13 problems. These problems test the exploration tendencies of the BMWOA and other metaheuristic algorithms. The average convergence patterns are also compared in logarithmic scale in Fig. 4.

Based on the results in Table 7, the overall rank of the BMWOA is one, followed by WOA, GWO, FA, BBO, GA, SCA, FPA, PSO, PBIL, and BAT. This algorithm also finds the best solutions for all problems compared to other techniques. In is evident that the WOA is among top three methods for these problems, while the proposed BMWOA obtains the most accurate results for F8–F13 functions. The foremost cause for the improved efficacy of the BMWOA is that it can make a well stable balance between the exploration and exploitation tendencies in dealing with multimodal landscapes. Hence, in the case of immature convergence, the BMWOA can effectively avoid converging to inferior positions. From Fig. 4, it is seen that the BMWOA has outperformed WOA and other peers in dealing with all cases.

Table 8 reveals results of the BMWOA versus other algorithms for F14–F23 test cases. All average convergence patterns are also exposed in Fig. 5. From results in Table 8, it can be seen that the BMWOA is the best overall technique in tackling F14–F23 cases, while the WOA, FPA, FA, GWO, BBO, PBIL, PSO, BAT, GA, and SCA are in the next places, respectively. The proposed BMWOA attains the best results in dealing with F14–F22 problems. Inspecting the curves in Fig. 5a–i, the BMWOA has attained better solutions with lower results and faster convergence behaviors compared to other approaches. Stagnation behaviors of FPA are obvious in dealing with F15 and F18, while other methods show a gradual accelerating trend for most of the cases. From Fig. 5b–e, it is

Table 8 The results for F14–F23 problems

Function	Metric	PBIL	GA	PSO	BBO	FPA	GWO	BAT	FA	SCA	WOA	BMWOA
F14	Avg	1.03114	0.998	1.0036	0.998	0.99802	4.0398	11.9077	4.2308	1.78725	1.19641	0.998
	Stdv	0.17843	5.6E−16	0.01955	5.6E−16	7.6E−05	3.6309	6.87411	2.85566	1.86394	2.43915	0.59523
	Rank	6	2	5	2	4	9	11	10	8	7	1
F15	Avg	0.0693	0.00613	0.00151	0.00671	0.00062	0.00168	0.00408	0.00162	0.00107	0.0005	0.00031
	Stdv	0.01403	0.00537	0.00033	0.00417	0.00012	0.005	0.00432	0.00355	0.00039	0.0004	0.00051
	Rank	11	9	5	10	3	7	8	6	4	2	1
F16	Avg	0	− 0.7762	− 1.0306	− 0.9261	− 1.0316	− 1.0316	− 0.7868	− 1.0316	− 1.0316	− 1.0316	− 1.0316
	Stdv	0	0.20844	0.00074	0.10941	7.4E−08	1.1E−08	0.37401	1.2E−09	4.9E−05	4.9E−14	2.1E−10
	Rank	11	10	7	8	5	4	9	3	6	2	1
F17	Avg	0.64453	0.46386	0.39922	0.48892	0.39789	0.39789	0.39789	0.39789	0.39891	0.39789	0.39789
	Stdv	0	0.0679	0.00049	0.02769	1.7E−11	6.3E−07	3.8E−10	2.8E−10	0.00068	1.6E−06	1.7E−12
	Rank	11	9	8	10	2	6	5	4	7	3	1
F18	Avg	3	3	3.05761	3	3	3.00001	11.1	3	3.00004	3	3
	Stdv	0	0	0.03052	0	5.7E−09	9.5E−06	21.0877	1E−08	8.8E−05	1.1E−05	8.6E−12
	Rank	1	1	10	1	6	8	11	7	9	5	1
F19	Avg	− 0.3348	− 3.5528	− 3.8603	− 3.7336	− 3.8628	− 3.8622	− 3.8628	− 3.8628	− 3.8555	− 3.8628	− 3.8628
	Stdv	1.7E−16	0.22432	0.0022	0.05781	1E−08	0.00163	4.5E−08	2.7E−10	0.00285	0.00355	1.5E−10
	Rank	11	10	7	9	3	6	4	2	8	5	1
F20	Avg	− 0.1657	− 2.2681	− 3.1673	− 3.0544	− 3.3005	− 3.2434	− 3.2824	− 3.261	− 2.9058	− 3.2544	− 3.322
	Stdv	2.8E−17	0.29622	0.01879	0.15581	0.0111	0.07931	0.05605	0.07173	0.31381	0.05909	0.08076
	Rank	11	10	7	8	2	6	3	4	9	5	1
F21	Avg	− 10.153	− 3.2405	− 4.3907	− 3.6789	− 10.146	− 9.4329	− 5.7134	− 6.1655	− 3.1248	− 10.153	− 10.153
	Stdv	5.3E−15	1.52385	1.41699	2.53943	0.00776	1.84554	3.07863	3.73015	1.83108	2.48752	1.6E−05
	Rank	2	10	8	9	4	5	7	6	11	3	1
F22	Avg	− 7.5681	− 4.4134	− 4.1424	− 10.403	− 10.33	− 10.402	− 4.7154	− 10.148	− 3.4805	− 10.403	− 10.403
	Stdv	2.65166	3.02189	1.68115	5.3E−15	0.09385	0.00076	2.94051	1.3734	1.69618	3.09723	2.3E−08
	Rank	7	9	10	2	5	4	8	6	11	3	1
F23	Avg	− 10.358	− 9.5644	− 6.0799	− 10.536	− 10.349	− 10.356	− 4.1001	− 10.281	− 4.4991	− 10.536	− 10.176
	Stdv	0.96227	2.20513	1.34073	3.6E−15	0.19854	0.96211	2.98982	1.37596	1.3073	3.20033	1.34897
	Rank	3	8	9	1	5	4	11	6	10	2	7
Sum		74	78	76	60	39	59	77	54	83	37	16
Overall		7	10	8	6	3	5	9	4	11	2	1

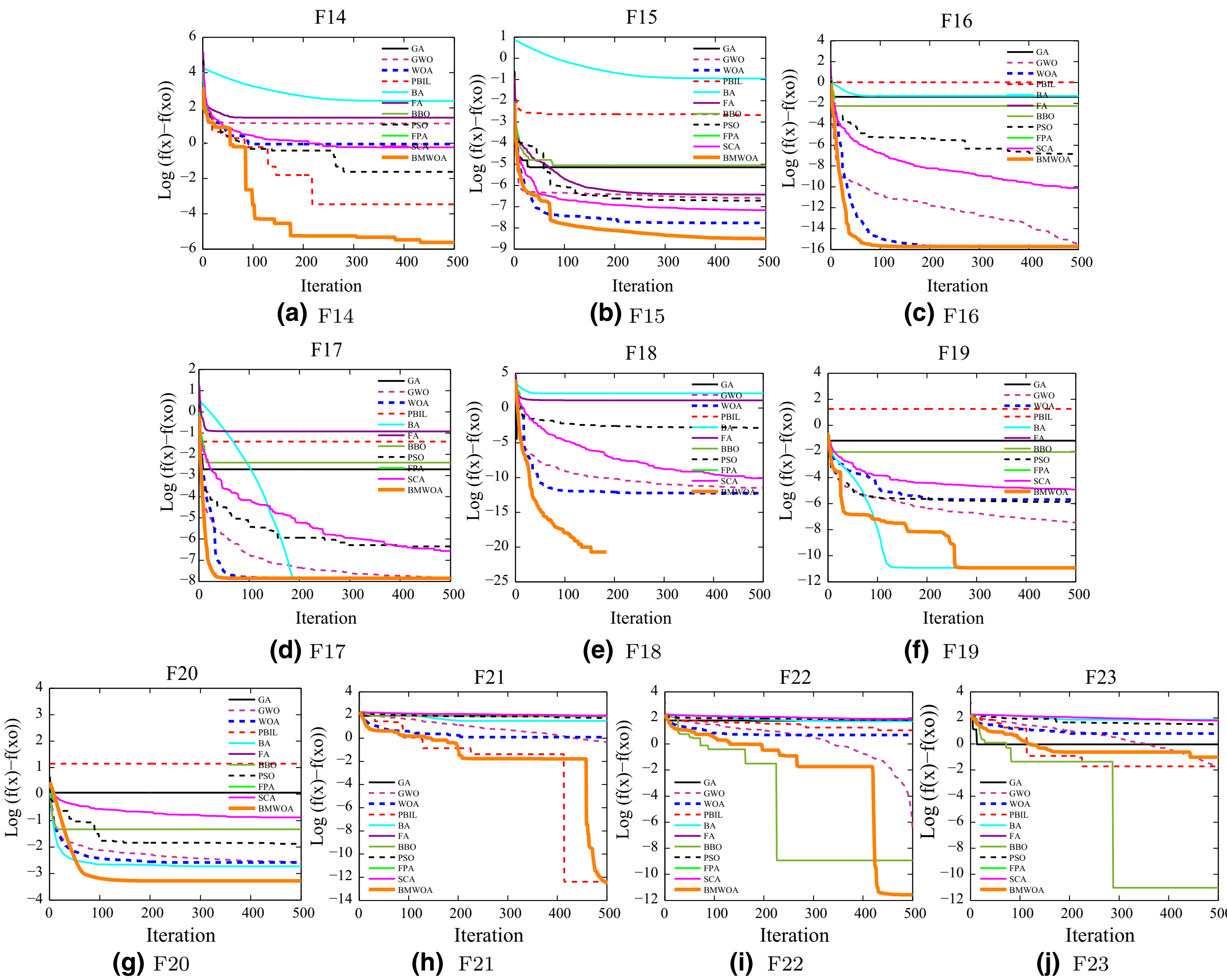

Fig. 5 Convergence curves for fixed dimensions functions (F14–F23)

observed that the WOA has shown the second best convergence behaviors.

From Table 9, it is verified that the observed differences in dealing with all F1–F23 are statistically significant for the majority of test cases according to p values of Wilcoxon rank-sum test with 5% significance. The observed improvements in multimodal landscapes are attributed to the ability to escape local solutions and avoid immature convergence of the BMWOA as a consequence of exploratory BHC-based local search mechanism with β-operator. Hence, in the case of stagnation shortcomings, it can search the neighborhood of the inferior solutions to escape from them. This fact can also be detected in the investigated convergence behaviors.

The obtained results for F1–F23 can reveal the improved diversification and intensification capacities of the developed BMWOA. In order to further prove the efficacy of the proposed optimizer in tackling more challenging tasks, it is applied to most recent CEC benchmark set for optimization purposes in Sect. 5.4.

5.4 Comparative results for CEC17 problems

In this part, the efficacy and convergence behaviors of the proposed BMWOA are evaluated based on the new IEEE CEC17 benchmark set [32]. Table 10 shows the details of the CEC17 problems. This set includes 30 bound constrained test cases with various characteristics and levels of complexity. It consists of the unimodal, multimodal, hybrid and composition functions with various optimum values in the range of [100, 3000] and can expose the exploration and exploitation leanings of the validated optimizers in realizing more complex and challenging landscapes. These problems are new test cases and still not utilized for the majority of optimizers including the SCA, WOA, and PBIL algorithms.

Table 9 The p values of the Wilcoxon test for BMWOA and other algorithms for F1–F23 problems ($p \geq 0.05$ are not significant and are underlined)

Benchmark	PBIL	GA	PSO	BBO	FPA	GWO	BAT	FA	SCA	WOA	BMWOA
F1	1.93E−11	1.95E−11	1.94E−11	1.95E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	NA
F2	3.89E−13	6.84E−13	6.84E−13	NA	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	NA
F3	2.24E−11	1.94E−11	1.95E−11	1.95E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	NA
F4	1.69E−11	1.94E−11	2.10E−11	1.95E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	NA
F5	2.23E−11	1.95E−11	1.95E−11	1.95E−11	3.02E−11	4.08E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	NA
F6	1.92E−11	1.95E−11	1.95E−11	1.94E−11	3.02E−11	3.69E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	NA
F7	2.06E−11	2.38E−11	1.95E−11	4.46E−03	3.02E−11	1.30E−01	3.02E−11	3.02E−11	3.02E−11	2.28E−01	NA
F8	1.90E−11	1.94E−11	1.94E−11	1.95E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	6.06E−11	NA
F9	6.73E−13	2.63E−12	6.83E−13	NA	1.21E−12	9.71E−12	1.21E−12	1.21E−12	1.20E−05	3.34E − 01	NA
F10	1.30E−11	1.32E−11	1.31E−11	1.31E−11	2.08E−11	1.51E−11	2.08E−11	2.08E−11	2.08E−11	2.08E−11	NA
F11	1.02E−11	1.02E−11	1.02E−11	1.02E−11	1.62E−11	5.62E − 02	1.62E-11	3.89E−05	1.62E−11	1.15E−03	NA
F12	1.88E−11	1.95E−11	1.94E−11	1.95E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	NA
F13	1.85E−11	1.95E−11	1.95E−11	2.08E−11	3.02E−11	3.02E−11	3.02E−11	8.20E−07	3.02E−11	4.50E−11	NA
F14	3.34E− 01	1.45E−02	6.83E−13	1.94E−09	4.20E−09	8.19E−04	7.35E−10	3.28E−11	4.38E−12	1.45E−02	NA
F15	5.18E−12	1.91E−11	1.29E−09	8.07E−10	6.76E−05	2.03E − 01	1.10E−08	5.85E−06	1.59E−07	1.09E−05	NA
F16	1.69E−14	8.22E−13	6.83E−13	6.66E−13	4.55E−12	1.63E−11	1.43E−04	6.55E−04	1.21E−12	3.34E − 01	NA
F17	1.69E−14	6.76E−13	6.83E−13	6.76E−13	2.70E−03	1.21E−12	2.70E−03	1.61E − 01	1.21E−12	1.66E−11	NA
F18	NA	NA	6.85E−13	NA	2.94E−07	1.21E−12	1.20E−12	1.62E−11	1.21E−12	1.21E−12	NA
F19	2.71E−14	9.77E−13	9.86E−13	1.40E−12	9.02E−12	1.72E−12	1.72E−12	3.01E−03	1.72E−12	1.72E−12	NA
F20	1.21E−12	1.94E−11	1.94E−11	1.41E−09	3.79E − 01	6.63E-01	2.71E−02	3.79E − 01	3.02E−11	8.77E − 01	NA
F21	2.75E−11	1.43E−11	1.62E−11	5.14E−12	2.54E−11	2.54E−11	6.25E−11	1.07E−10	2.54E−11	2.54E−11	NA
F22	1.18E−11	1.27E−11	1.59E−11	9.38E−13	2.48E−11	2.48E−11	2.48E−11	2.60E−11	2.48E−11	2.48E−11	NA
F23	3.34E − 01	2.15E−02	6.85E−13	NA	1.21E−12	1.21E−12	2.21E−06	3.35E−11	1.21E−12	1.21E−12	6.84E−10

Table 10 CEC2017 test suite

ID	Name of the function	Class	Optimum
C01	Shifted and rotated bent cigar function	Unimodal	100
C02	Shifted and rotated sum of different power function	Unimodal	200
C03	Shifted and rotated Zakharov function	Unimodal	300
C04	Shifted and rotated Rosenbrock's function	Multimodal	400
C05	Shifted and rotated Rastrigin's function	Multimodal	500
C06	Shifted and rotated expanded Scaffer's F6 function	Multimodal	600
C07	Shifted and rotated Lunacek Bi_Rastrigin function	Multimodal	700
C08	Shifted and rotated non-continuous Rastrigin's function	Multimodal	800
C09	Shifted and rotated Levy function	Multimodal	900
C10	Shifted and Rotated Schwefel's Function	Multimodal	1000
C11	Hybrid function 1 ($N = 3$)	Hybrid	1100
C12	Hybrid function 2 ($N = 3$)	Hybrid	1200
C13	Hybrid function 3 ($N = 3$)	Hybrid	1300
C14	Hybrid function 4 ($N = 4$)	Hybrid	1400
C15	Hybrid function 5 ($N = 4$)	Hybrid	1500
C16	Hybrid function 6 ($N = 4$)	Hybrid	1600
C17	Hybrid function 6 ($N = 5$)	Hybrid	1700
C18	Hybrid function 6 ($N = 5$)	Hybrid	1800
C19	Hybrid function 6 ($N = 5$)	Hybrid	1900
C20	Hybrid function 6 ($N = 6$)	Hybrid	2000
C21	Composition function 1 ($N = 3$)	Composition	2100
C22	Composition function 2 ($N = 3$)	Composition	2200
C23	Composition function 3 ($N = 4$)	Composition	2300
C24	Composition function 4 ($N = 4$)	Composition	2400
C25	Composition function 5 ($N = 5$)	Composition	2500
C26	Composition function 6 ($N = 5$)	Composition	2600
C27	Composition function 7 ($N = 6$)	Composition	2700
C28	Composition function 8 ($N = 6$)	Composition	2800
C29	Composition function 9 ($N = 3$)	Composition	2900
C30	Composition function 10 ($N = 3$)	Composition	3000

The proposed BMWOA technique is compared to a series of well-established and new algorithms including the GA [47], PSO [47], PBIL [47], BAT [51], SCA [38], and WOA [39] optimizers. All tests are performed based on the recommendations of the IEEE CEC2017 report, and all algorithms are tested in the same testing environment similar to previous experiments to form a fair judgment. The parameters are same as those reported in Table 5 from previous tests.

The dimension of CEC17 benchmarks was set to 30 and their variables can change inside interval [− 100, 100] for all problems. The maximum function evaluations of all methods were set to 3.00E+05 as recommended in CEC2017.

Tables 11, 12, 13 and 14 present the obtained metrics of the proposed BMWOA compared to other assessed algorithms based on normalized error rates in dealing with C1 to C30 problems. The convergence trends are also compared in Figs. 6, 7, 8 and 9.

As per results in Table 11, the BMWOA has attained the best overall rank on C1–C6, while the GA, SCA, WOA, PBIL, BAT, and PSO are placed in next stages, respectively. For C1, the best methods are BMWOA, WOA, GA, SCA, BAT, PSO, and PBIL, respectively. For C2, the best choices are BMWOA, GA, WOA, SCA, PBIL, PSO, and BAT, respectively. For all unimodal C1–C3 problems, the proposed BMWOA has reached to the lowest error rates, and it is observed that the worst results of BMWOA are much better than the best solutions of all other algorithms. This indicates that the exploitation tendency of the WOA on unimodal landscapes has improved, considerably. The reason is that the BMWOA can put more emphasis on exploitative patterns in later steps because of the embedded BHC-based local search operators plus the exploitation factors of the WOA. Hence, it finds better solutions, while

Table 11 Results for C1–C6 problems

ID	Metric	PSO	GA	PBIL	BAT	SCA	WOA	BMWOA
C1	Avg	4.76E−01	8.20E−05	8.09E−01	1.00E+00	2.20E−01	1.27E−05	0.00E+00
	Best	4.14E−01	1.25E−04	9.80E−01	1.00E+00	2.31E−01	2.33E−06	0.00E+00
	Worst	4.43E−01	6.09E−05	6.34E−01	1.00E+00	2.11E−01	3.44E−05	0.00E+00
	Stdv	4.10E−01	7.19E−06	4.13E−01	1.00E+00	1.79E−01	6.41E−05	0.00E+00
	Rank	5	3	6	5	4	2	1
C2	Avg	2.82E−06	1.32E−32	1.27E−06	1.00E+00	1.32E−07	1.95E−22	0.00E+00
	Best	1.00E+00	5.09E−29	2.34E−02	1.23E−02	4.28E−06	1.41E−19	0.00E+00
	Worst	2.54E−06	1.16E−32	1.57E−06	1.00E+00	2.37E−07	3.47E−22	0.00E+00
	Stdv	2.34E−06	9.99E−33	1.32E−06	1.00E+00	1.90E−07	2.79E−22	0.00E+00
	Rank	6	2	5	6	4	3	1
C3	Avg	6.19E−01	2.14E−01	5.40E−01	1.00E+00	2.08E−01	8.87E−01	0.00E+00
	Best	1.00E+00	2.80E−01	7.10E−01	6.20E−01	2.47E−01	6.12E−01	0.00E+00
	Worst	3.68E−01	1.74E−01	3.21E−01	1.00E+00	1.31E−01	7.19E−01	0.00E+00
	Stdv	2.39E−01	1.21E−01	1.72E−01	9.11E−01	9.31E−02	1.00E+00	0.00E+00
	Rank	5	3	4	5	2	6	1
C4	Avg	3.55E−01	9.80E−04	2.36E−01	1.00E+00	8.03E−02	3.08E−03	0.00E+00
	Best	7.65E−01	0.00E+00	5.39E−01	1.00E+00	1.72E−01	1.83E−02	1.07E−02
	Worst	3.11E−01	5.78E−03	1.61E−01	1.00E+00	7.36E−02	4.10E−03	0.00E+00
	Stdv	1.79E−01	2.82E−03	7.87E−02	1.00E+00	3.65E−02	2.10E−03	0.00E+00
	Rank	6	2	5	6	4	3	1
C5	Avg	7.90E−01	0.00E+00	7.82E−01	1.00E+00	4.80E−01	5.03E−01	3.35E−01
	Best	8.08E−01	0.00E+00	1.00E+00	9.88E−01	5.47E−01	3.60E−01	2.29E−01
	Worst	6.28E−01	0.00E+00	5.44E−01	1.00E+00	3.74E−01	7.52E−01	6.05E−01
	Stdv	2.51E−01	2.35E−01	0.00E+00	7.70E−01	1.70E−01	1.00E+00	9.36E−01
	Rank	6	1	5	6	3	4	2
C6	Avg	8.91E−01	0.00E+00	1.00E+00	9.42E−01	5.66E−01	8.67E−01	6.69E−01
	Best	8.90E−01	0.00E+00	1.00E+00	8.46E−01	5.27E−01	5.74E−01	3.17E−01
	Worst	7.60E−01	0.00E+00	8.13E−01	9.76E−01	5.46E−01	1.00E+00	7.32E−01
	Stdv	3.31E−01	0.00E+00	4.18E−01	7.68E−01	4.23E−01	1.00E+00	9.26E−01
	Rank	5	1	7	5	2	4	3
$\sum$	Sum	33	12	32	33	19	22	9
	Overall	6	2	5	6	3	4	1

the other optimizers cannot make a more stable balance between the global and local search operations. In the case of any local optima stagnation, controlled exploratory jumps around the vicinity of solutions enhance the exploitation capacity of the method. The impact of this mechanism can be vividly seen in curves for C01–C03 cases in Fig. 6.

For multimodal C4, the best algorithm is the BMWOA, while the GA and WOA are in the subsequent places. The SCA, PBIL, PSO, and BAT also attain the next ranks, respectively.

For C5 and C6, the BMWOA can realize enriched results compared to the original WOA, BAT, PSO, and PBIL. The reason is that the BMWOA can establish a more stable balance between exploration and exploitation leanings on multimodal cases since the new learning and local search operators can assist it to smoothly shift from the improved exploratory patterns in first steps to exploitation in later phases.

From the convergence tendencies in Fig. 6, an accelerated and superior performance can be easily seen in the curves of BMWOA on C1–C3 cases (see Fig. 6a–c). As it was observed in Table 11, the curves of GA and WOA on C4 are competitive with BMWOA, while the GA outperforms others on C5 and C6.

From Table 12, it is observed that the BMWOA is always between three top techniques when solving multimodal C7, C8, C9, C10, and hybrid C12 problems, while for hybrid C11, C13, C14, and C15 cases, it evidently outperforms the standard WOA and other optimizers. For all C7–C15 problems in Table 12, the BMWOA outperforms the WOA, BAT, PBIL, and PSO algorithms

Table 12 Results for C7–C15 problems

ID	Metric	PSO	GA	PBIL	BAT	SCA	WOA	BMWOA
C7	Avg	2.68E−01	0.00E+00	5.64E−01	1.00E+00	1.02E−01	1.50E−01	1.15E−01
	Best	2.62E−01	0.00E+00	5.73E−01	1.00E+00	1.18E−01	1.06E−01	7.40E−02
	Worst	2.54E−01	0.00E+00	4.70E−01	1.00E+00	8.21E−02	1.62E−01	1.22E−01
	Stdv	2.09E−01	0.00E+00	1.62E−01	1.00E+00	1.52E−02	2.78E−01	2.27E−01
	Rank	5	1	6	5	2	4	3
C8	Avg	8.62E−01	0.00E+00	1.00E+00	9.24E−01	5.16E−01	3.57E−01	2.88E−01
	Best	9.47E−01	0.00E+00	1.00E+00	5.78E−01	5.63E−01	1.90E−01	9.93E−02
	Worst	6.89E−01	0.00E+00	8.60E−01	1.00E+00	3.83E−01	7.23E−01	3.55E−01
	Stdv	1.14E−01	2.13E−01	0.00E+00	7.30E−01	7.65E−02	1.00E+00	5.45E−01
	Rank	5	1	7	5	4	3	2
C9	Avg	7.64E−01	0.00E+00	1.00E+00	8.31E−01	3.49E−01	6.76E−01	4.57E−01
	Best	6.26E−01	0.00E+00	1.00E+00	6.30E−01	2.66E−01	3.63E−01	2.83E−01
	Worst	8.17E−01	0.00E+00	7.69E−01	8.85E−01	2.61E−01	1.00E+00	5.31E−01
	Stdv	4.76E−01	0.00E+00	1.24E−01	4.36E−01	1.35E−01	1.00E+00	4.91E−01
	Rank	5	1	7	5	2	4	3
C10	Avg	1.00E+00	0.00E+00	9.93E−01	4.25E−01	9.66E−01	4.63E−01	3.43E−01
	Best	9.51E−01	0.00E+00	1.00E+00	2.42E−01	8.98E−01	1.55E−01	1.65E−01
	Worst	1.00E+00	0.00E+00	9.81E−01	5.06E−01	9.88E−01	6.64E−01	4.50E−01
	Stdv	1.08E−01	4.23E−01	0.00E+00	5.00E−01	1.64E−01	1.00E+00	5.90E−01
	Rank	7	1	6	7	5	4	2
C11	Avg	2.85E−01	4.90E−02	2.75E−01	1.00E+00	5.29E−02	7.02E−03	0.00E+00
	Best	6.31E−01	5.39E−02	1.00E+00	3.32E−01	1.68E−01	1.21E−02	0.00E+00
	Worst	2.07E−01	1.53E−01	1.61E−01	1.00E+00	4.41E−02	4.12E−03	0.00E+00
	Stdv	1.38E−01	1.35E−01	4.18E−02	1.00E+00	2.47E−02	9.36E−04	0.00E+00
	Rank	6	3	5	6	4	2	1
C12	Avg	4.10E−01	0.00E+00	6.19E−01	1.00E+00	1.65E−01	3.46E−03	2.32E−04
	Best	3.77E−01	5.46E−04	1.00E+00	3.24E−01	2.01E−01	7.43E−04	0.00E+00
	Worst	3.26E−01	0.00E+00	3.54E−01	1.00E+00	1.23E−01	4.09E−03	5.35E−05
	Stdv	2.47E−01	0.00E+00	1.89E−01	1.00E+00	8.79E−02	5.29E−03	4.05E−04
	Rank	5	1	6	5	4	3	2
C13	Avg	7.04E−01	9.35E−04	1.00E+00	9.88E−01	3.32E−01	7.55E−05	0.00E+00
	Best	5.28E−01	9.23E−04	1.00E+00	7.06E−05	4.79E−01	1.54E−05	0.00E+00
	Worst	2.30E−01	3.00E−04	3.18E−01	1.00E+00	1.20E−01	3.87E−05	0.00E+00
	Stdv	2.26E−01	2.30E−04	3.36E−01	1.00E+00	1.04E−01	3.57E−05	0.00E+00
	Rank	5	3	7	5	4	2	1
C14	Avg	2.29E−01	5.15E−01	3.85E−01	1.00E+00	5.99E−02	5.76E−01	0.00E+00
	Best	7.25E−01	1.00E+00	9.22E−01	9.38E−03	3.38E−01	1.41E−01	0.00E+00
	Worst	5.82E−02	4.34E−01	9.88E−02	1.00E+00	1.55E−02	5.81E−01	0.00E+00
	Stdv	3.98E−02	3.51E−01	5.45E−02	1.00E+00	5.02E−03	4.64E−01	0.00E+00
	Rank	3	5	4	3	2	6	1
C15	Avg	4.66E−01	4.13E−03	1.00E+00	2.85E−02	6.16E−02	4.26E−04	0.00E+00
	Best	4.52E−01	3.63E−03	1.00E+00	1.53E−04	7.36E−03	2.77E−04	0.00E+00
	Worst	5.22E−01	3.73E−03	1.00E+00	4.42E−01	1.56E−01	1.40E−03	0.00E+00
	Stdv	5.37E−01	4.37E−03	1.00E+00	4.55E−01	1.73E−01	1.55E−03	0.00E+00
	Rank	6	3	7	6	5	2	1
$\sum$	Sum	47	19	55	47	32	30	16
	Overall	5	2	7	5	4	3	1

Table 13 Results for C16–C19 problems

ID	Metric	PSO	GA	PBIL	BAT	SCA	WOA	BMWOA
C16	Avg	7.91E−01	2.88E−02	7.15E−01	1.00E+00	4.57E−01	4.53E−01	0.00E+00
	Best	9.89E−01	9.13E−02	1.00E+00	6.01E−01	6.92E−01	4.12E−01	0.00E+00
	Worst	3.77E−01	0.00E+00	3.26E−01	1.00E+00	2.33E−01	4.62E−01	1.06E−01
	Stdv	6.09E−02	1.15E−01	0.00E+00	1.00E+00	1.01E−01	4.21E−01	3.69E−01
	Rank	6	2	5	7	4	3	1
C17	Avg	1.93E−01	6.84E−03	2.27E−01	1.00E+00	5.04E−02	9.39E−02	0.00E+00
	Best	4.84E−01	3.54E−02	4.95E−01	1.00E+00	1.60E−01	7.19E−02	0.00E+00
	Worst	1.53E−01	1.46E−02	1.48E−01	1.00E+00	2.90E−02	1.24E−01	0.00E+00
	Stdv	4.40E−02	4.31E−02	0.00E+00	1.00E+00	4.19E−02	2.79E−01	8.99E−02
	Rank	5	2	6	7	3	4	1
C18	Avg	3.70E−01	8.11E−02	3.68E−01	1.00E+00	1.50E−01	1.06E−01	0.00E+00
	Best	1.00E+00	8.08E−02	8.24E−01	0.00E+00	5.15E−01	6.92E−02	1.61E−03
	Worst	1.01E−01	7.25E−02	6.96E−02	1.00E+00	4.01E−02	5.25E−02	0.00E+00
	Stdv	6.52E−02	6.98E−02	4.78E−02	1.00E+00	2.72E−02	5.50E−02	0.00E+00
	Rank	6	2	5	7	4	3	1
C19	Avg	3.69E−01	3.73E−03	1.00E+00	2.42E−02	9.62E−02	8.59E−03	0.00E+00
	Best	1.15E−01	3.19E−03	1.00E+00	2.85E−04	5.35E−02	7.20E−04	0.00E+00
	Worst	5.83E−01	3.53E−03	1.00E+00	1.03E−01	1.18E−01	1.21E−02	0.00E+00
	Stdv	5.43E−01	2.88E−03	1.00E+00	1.03E−01	1.30E−01	1.52E−02	0.00E+00
	Rank	6	2	7	4	5	3	1
	Sum	23	8	23	25	16	13	4
	Overall	5	2	5	7	4	3	1

considering all metrics. In addition, superior convergence trends in dealing with C11, C13–C15 can be seen in Fig. 7. The BMWOA has an improved BHC-based local search when it finds a promising solution; therefore, it can show accelerated and competitive convergence curves.

For the hybrid test cases (C16–C19) in Table 13, the BMWOA is capable of finding the best solutions, and based on the total ranks, it outperforms the GA, WOA, SCA, PBIL, PSO, and BA optimizers, respectively. For C16, BMWOA has attained the first rank followed by GA, WOA, SCA, PBIL, PSO, and BAT. The results for hybrid problems reveal that the BAT, PBIL, and PSO have not preserved a fine balance between exploration and exploitation trends. On the other hand, methods such as BMWOA, WOA, and GA have shown a more stable balance between the core searching trends, which has led to a better overall performance on these hybrid cases. In the case of immature convergence and entrapment in local solutions, the BMWOA has escaped from them because of the role of random leaders and it also utilizes associative learning theme and STM in addition to an exploratory local β-operator in last steps. Regarding the convergences in Fig. 8, a similar pattern can be seen in the curves and the BMWOA shows superior accelerations compared to other competitors. The S-operator can retain the enhanced whales throughout the BHC-based phase; hence, the quality of results and speed of convergence have improved in the BMWOA.

Table 14 shows that the BMWOA algorithm provides the best results on C24, C25, C27, C28, and C30 problems. It is the second best strategy on hybrid C20, composition C21–C23, C26, and C29. We see that the GA also shows a stable performance on C20–C23, C26, and C29 cases with obtaining best results. In addition, the BAT and PSO cannot show sufficient exploratory and exploitative powers in dealing with composition cases; therefore, they cannot obtain superior overall results on C20, C21, C23, C24, and C26–C29 problems. In all composition cases, the proposed WOA-based optimizer outperforms the WOA method. The reason is that the BMWOA utilizes an exploratory BHC-based local search with β-operator in later steps in addition to its enhanced exploratory behaviors using random leaders, associative learning, and STM in initial iterations; hence, it can make a fine balance amid its exploratory and exploitative inclinations and discloses an enriched potential in effectively escaping from the inferior solutions.

Table 14 Results for C20–C30 problems

ID	Metric	PSO	GA	PBIL	BAT	SCA	WOA	BMWOA
C20	Avg	4.62E−01	0.00E+00	3.83E−01	1.00E+00	2.78E−01	3.91E−01	1.62E−01
	Best	7.57E−01	8.02E−02	8.33E−01	1.00E+00	3.52E−01	2.44E−01	0.00E+00
	Worst	2.19E−01	0.00E+00	1.69E−01	1.00E+00	2.53E−01	3.57E−01	1.35E−01
	Stdv	1.09E−01	1.45E−01	0.00E+00	1.00E+00	2.39E−01	6.50E−01	4.27E−01
	Rank	6	1	4	7	3	5	2
C21	Avg	7.57E−01	0.00E+00	7.91E−01	1.00E+00	4.24E−01	6.63E−01	2.98E−01
	Best	8.48E−01	0.00E+00	1.00E+00	5.76E−01	5.50E−01	3.11E−01	1.89E−01
	Worst	5.33E−01	0.00E+00	4.28E−01	1.00E+00	1.87E−01	8.37E−01	3.05E−01
	Stdv	1.54E−01	1.47E−01	0.00E+00	9.53E−01	8.08E−02	1.00E+00	4.98E−01
	Rank	5	1	6	7	3	4	2
C22	Avg	7.22E−01	0.00E+00	4.10E−01	7.93E−01	1.00E+00	2.51E−01	1.07E−01
	Best	5.85E−01	1.20E−02	8.06E−01	1.00E+00	2.82E−01	2.06E−03	0.00E+00
	Worst	6.48E−01	0.00E+00	5.51E−01	1.00E+00	6.16E−01	4.46E−01	4.10E−01
	Stdv	8.93E−01	0.00E+00	6.55E−01	8.55E−02	1.00E+00	9.37E−01	9.75E−01
	Rank	5	1	4	6	7	3	2
C23	Avg	3.96E−01	0.00E+00	1.92E−01	1.00E+00	2.19E−01	2.80E−01	1.58E−01
	Best	3.06E−01	0.00E+00	4.28E−01	1.00E+00	3.04E−01	9.30E−02	6.82E−02
	Worst	3.35E−01	0.00E+00	1.14E−01	1.00E+00	1.46E−01	3.54E−01	2.36E−01
	Stdv	2.91E−01	8.24E−02	0.00E+00	1.00E+00	8.48E−02	5.46E−01	4.02E−01
	Rank	6	1	3	7	4	5	2
C24	Avg	3.14E−01	2.98E−02	2.84E−02	1.00E+00	1.21E−01	1.83E−01	0.00E+00
	Best	4.25E−01	1.39E−01	2.16E−01	1.00E+00	3.06E−01	1.03E−01	0.00E+00
	Worst	2.77E−01	6.35E−02	0.00E+00	1.00E+00	8.80E−02	2.83E−01	1.31E−01
	Stdv	2.75E−01	4.57E−01	0.00E+00	1.00E+00	7.36E−02	6.12E−01	5.45E−01
	Rank	6	3	2	7	4	5	1
C25	Avg	4.51E−01	1.16E−02	9.47E−01	1.00E+00	1.00E−01	8.97E−03	0.00E+00
	Best	4.99E−01	7.79E−03	1.00E+00	7.58E−01	1.08E−01	4.18E−03	0.00E+00
	Worst	3.59E−01	2.20E−02	6.78E−01	1.00E+00	8.24E−02	2.89E−03	0.00E+00
	Stdv	2.73E−01	1.27E−02	4.28E−01	1.00E+00	4.09E−02	6.20E−03	0.00E+00
	Rank	5	3	6	7	4	2	1
C26	Avg	4.77E−01	0.00E+00	3.31E−01	1.00E+00	2.70E−01	3.29E−01	1.83E−01
	Best	5.06E−01	2.33E−02	6.58E−01	1.00E+00	5.68E−01	8.08E−02	0.00E+00
	Worst	3.82E−01	0.00E+00	2.06E−01	1.00E+00	1.68E−01	4.01E−01	3.48E−01
	Stdv	3.77E−01	2.53E−01	0.00E+00	9.15E−01	1.03E−01	1.00E+00	7.62E−01
	Rank	6	1	5	7	3	4	2
C27	Avg	1.88E−01	5.91E−03	3.78E−02	1.00E+00	1.04E−01	6.07E−02	0.00E+00
	Best	1.28E−01	3.80E−02	1.20E−01	1.00E+00	1.36E−01	6.48E−02	0.00E+00
	Worst	1.27E−01	0.00E+00	2.11E−02	1.00E+00	5.80E−02	8.67E−02	3.67E−02
	Stdv	1.15E−01	2.04E−02	0.00E+00	1.00E+00	6.29E−02	1.30E−01	7.29E−02
	Rank	6	2	3	7	5	4	1
C28	Avg	5.14E−01	1.88E−02	3.63E−01	1.00E+00	1.69E−01	1.92E−02	0.00E+00
	Best	8.69E−01	3.20E−02	6.89E−01	1.00E+00	2.85E−01	3.06E−02	0.00E+00
	Worst	5.03E−01	1.43E−02	3.70E−01	1.00E+00	1.48E−01	1.85E−02	0.00E+00
	Stdv	2.68E−01	0.00E+00	1.56E−01	1.00E+00	7.62E−02	1.99E−04	1.64E−03
	Rank	6	2	5	7	4	3	1

Table 14 (continued)

ID	Metric	PSO	GA	PBIL	BAT	SCA	WOA	BMWOA
C29	Avg	2.82E−01	0.00E+00	2.03E−01	1.00E+00	1.95E−01	2.65E−01	7.34E−02
	Best	4.28E−01	0.00E+00	3.88E−01	1.00E+00	3.28E−01	2.26E−01	8.11E−02
	Worst	1.36E−01	0.00E+00	1.04E−01	1.00E+00	8.88E−02	2.22E−01	8.28E−02
	Stdv	4.36E−02	1.76E−02	0.00E+00	1.00E+00	1.68E−02	2.32E−01	1.25E−01
	Rank	6	1	4	7	3	5	2
C30	Avg	5.89E−01	1.04E−03	1.00E+00	9.88E−01	3.44E−01	3.48E−02	0.00E+00
	Best	5.48E−01	3.52E−04	1.00E+00	4.08E−02	4.25E−01	3.71E−03	0.00E+00
	Worst	2.06E−01	0.00E+00	2.46E−01	1.00E+00	1.06E−01	3.90E−02	6.18E−04
	Stdv	2.31E−01	0.00E+00	1.39E−01	1.00E+00	8.06E−02	2.99E−02	4.35E−04
	Rank	5	2	7	6	4	3	1
	Sum	62	18	49	75	44	43	17
	Overall	6	2	5	7	4	3	1

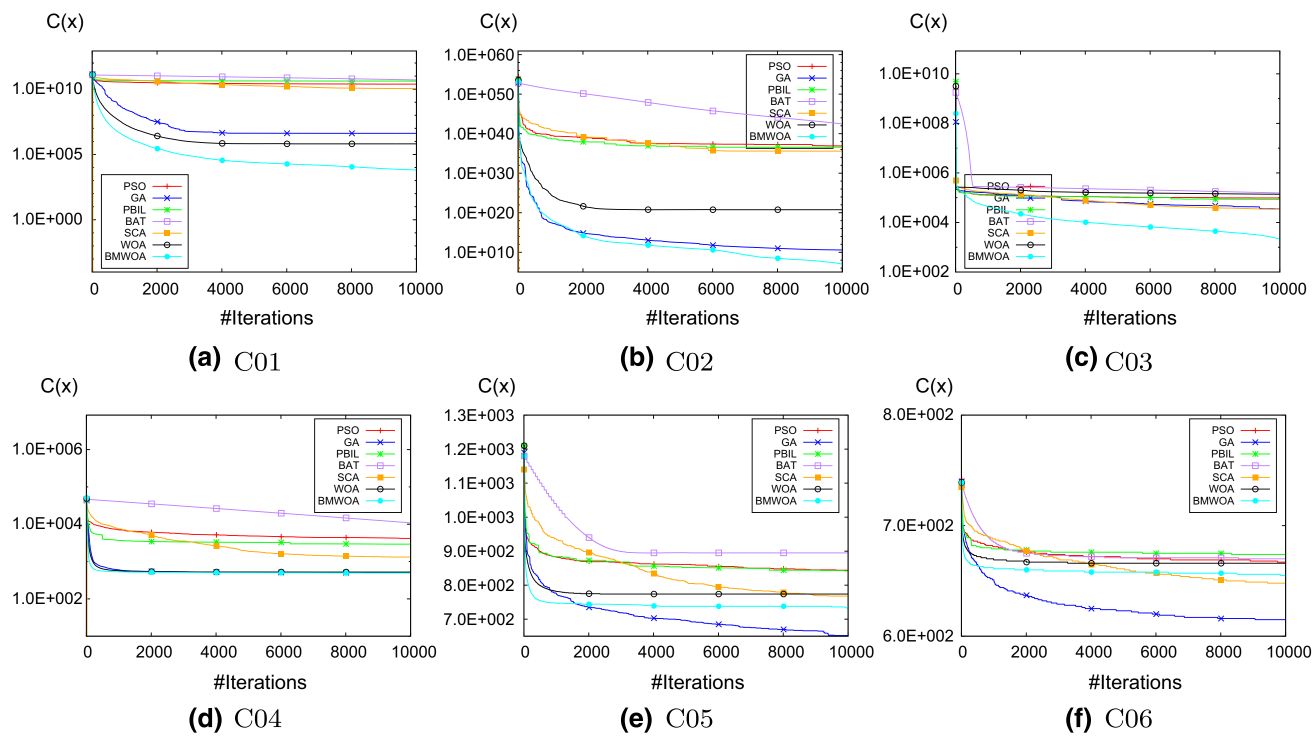

Fig. 6 Convergence curves for C01–C06 problems

Compared to WOA, BMWOA also can exhibit matching exploitative and exploratory behaviors on some steps based on embedded random switching parameters. For a series of steps, the BMWOA can switch to some alternative searching strategies having the proposed global and local search operators. Thus, in comparison with the standard WOA, the BMWOA has revealed relatively superior results on all composition functions. In total, it attains the best rank for the functions in Table 14, followed by GA, WOA, SCA, PBIL, PSO, and BAT optimizers. Inspecting the results in Fig. 9, the developed BMWOA can reveal a faster convergence behavior on C24, C25, C27, C28, and C30 problems. All convergence patterns in Fig. 9 are in accordance with results in Table 14. The foremost cause of accelerated curves for BMWOA is due to this trait that the *S*-operator only retains improved whales during the BHC-

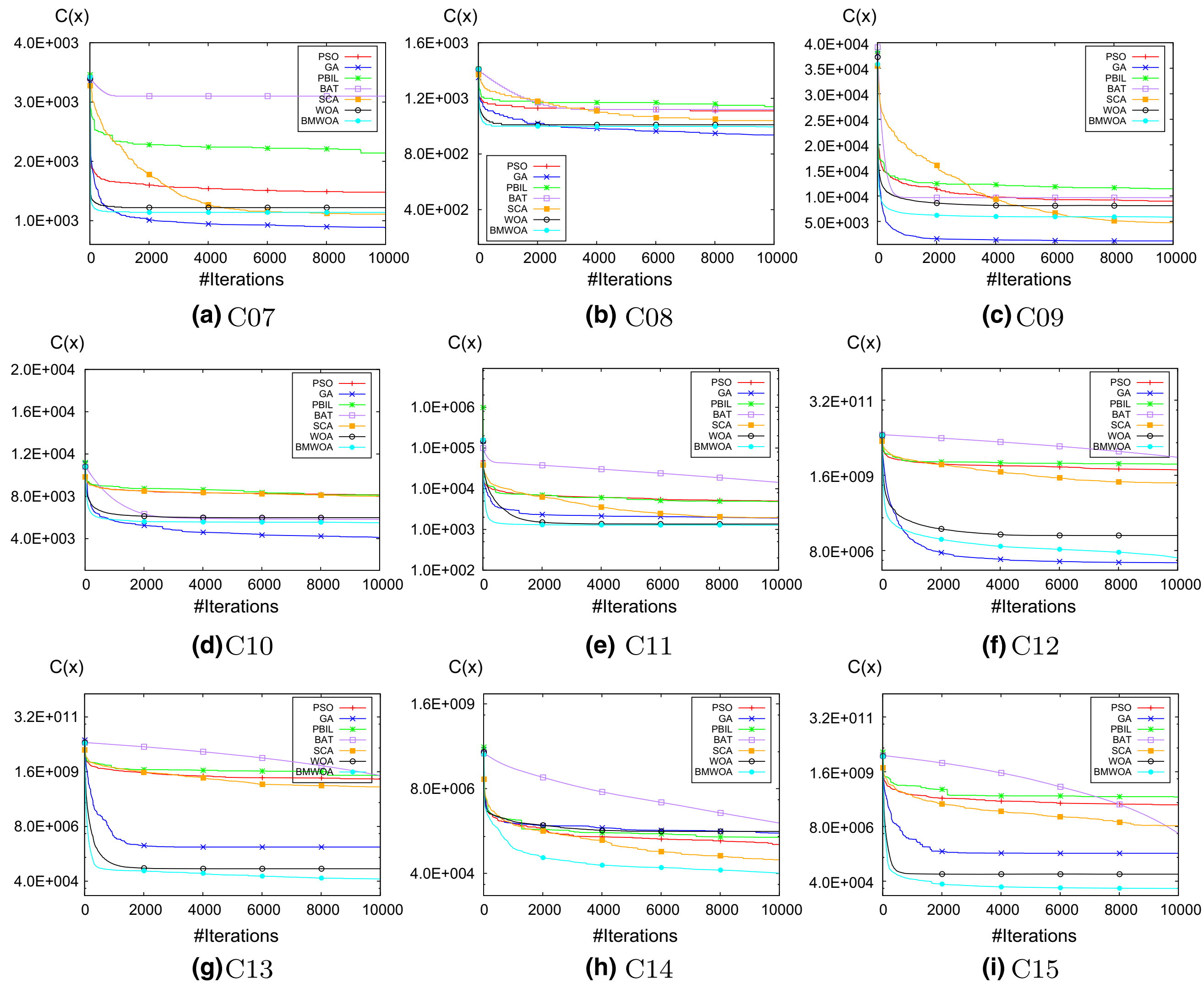

Fig. 7 Convergence curves for C07–C15 problems

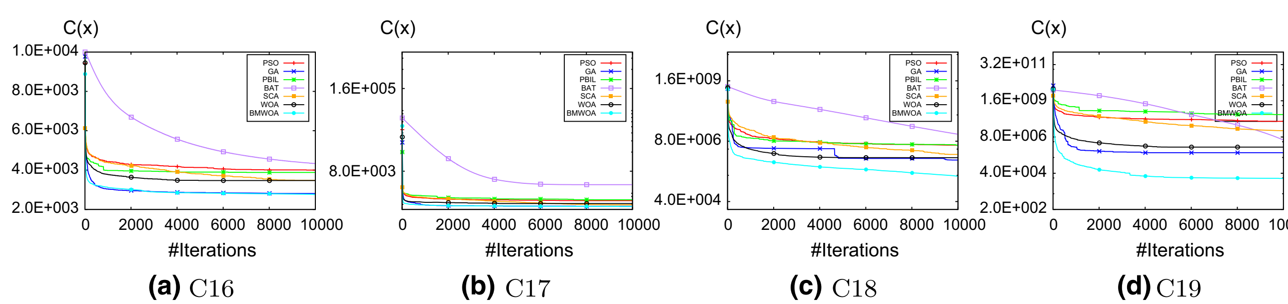

Fig. 8 Convergence curves for C16–C19 problems

based exploitative steps and after improved exploration at beginning steps and finding preferable solutions. Consequently, it can relatively converge faster and earlier than WOA.

Based on the p values in Table 15, it is seen that the observed performance improvements of BMWOA compared to other algorithms are statistically significant for the majority of test cases.

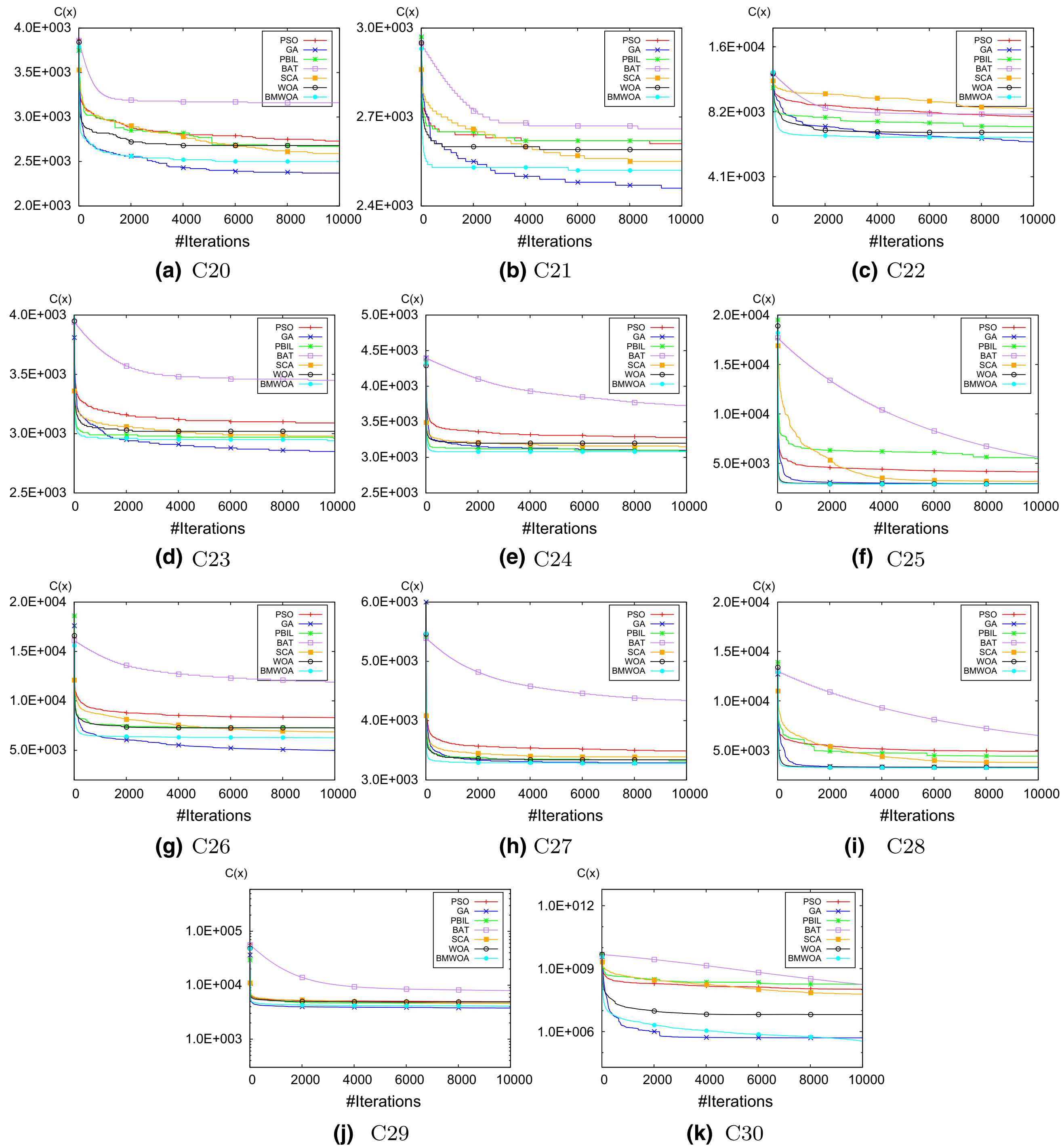

Fig. 9 Convergence curves for C20–C30 problems

6 Conclusions and future directions

In this paper, an enhanced WOA algorithm was proposed with a modified global searching operator to mitigate the immature convergence of the WOA algorithm and tackle different optimization challenges more efficiently. The proposed BMWOA algorithm utilizes the BHC local search engine during the exploitation phase to further exploit the neighborhood of the downhill whales. The BHC can assist the WOA in showing exploratory local search propensities in case of stagnation in local optima. The proposed approach was tested using different sets of benchmark problems and compared to several well-regarded meta-heuristic algorithms. The comprehensive experimental

Table 15 The p values of the Wilcoxon test for BMWOA and other algorithms in solving C01–C30 problems ($p \geq 0.05$ are underlined)

Benchmark	PSO	GA	PBIL	BAT	SCA	WOA	BMWOA
C01	3.02E−11	1.95E−11	1.95E−11	3.02E−11	3.02E−11	3.02E−11	NA
C02	3.02E−11	3.25E−09	1.96E−11	3.02E−11	3.02E−11	3.02E−11	NA
C03	3.02E−11	2.51E−11	1.96E−11	3.02E−11	3.02E−11	3.02E−11	NA
C04	3.02E−11	2.13E−01	1.95E−11	3.02E−11	3.02E−11	4.35E−05	NA
C05	2.50E−11	NA	2.15E−11	2.50E−11	2.50E−11	4.14E−11	8.71E−08
C06	2.66E−11	NA	2.20E−11	2.66E−11	2.66E−11	2.66E−11	2.66E−11
C07	1.96E−11	NA	1.24E−11	1.96E−11	1.96E−11	1.96E−11	1.96E−11
C08	1.96E−11	NA	1.25E−11	1.96E−11	1.96E−11	1.47E−08	2.70E−08
C09	1.95E−11	NA	1.67E−11	1.95E−11	3.98E−11	2.16E−11	2.65E−11
C10	1.96E−11	NA	1.72E−11	8.01E−11	1.96E−11	3.46E−10	1.72E−09
C11	3.02E−11	9.82E−10	1.95E−11	3.02E−11	3.02E−11	1.75E−05	NA
C12	2.26E−11	NA	1.68E−11	2.26E−11	2.26E−11	1.16E−08	1.00E − 01
C13	3.02E−11	1.95E−11	2.60E−11	9.26E−09	3.02E−11	6.05E−07	NA
C14	9.92E−11	1.28E−10	3.25E−11	1.86E−03	3.16E−05	1.29E−09	NA
C15	3.02E−11	2.72E−11	1.95E−11	1.10E−08	3.02E−11	1.43E−08	NA
C16	3.02E−11	6.08E−01	1.95E−11	7.38E−10	5.97E−09	4.44E−07	NA
C17	3.02E−11	9.11E − 01	1.95E−11	3.02E−11	2.50E−03	9.52E−04	NA
C18	3.02E−11	1.60E−06	2.17E−11	2.00E−05	8.15E−11	2.77E−05	NA
C19	3.02E−11	1.96E−11	2.39E−11	8.15E−11	3.02E−11	3.34E−11	NA
C20	8.01E−11	NA	1.34E−10	2.17E−11	2.70E−08	2.27E−08	7.63E−04
C21	1.95E−11	NA	1.53E−11	2.40E−11	5.38E−11	5.38E−11	1.35E−08
C22	5.89E−03	NA	8.51E−01	8.01E−11	5.49E−06	1.37E−04	2.16E−03
C23	2.26E−11	NA	2.03E−11	2.26E−11	2.26E−11	2.95E−10	1.01E−09
C24	6.07E−11	4.02E−01	1.82E−01	3.02E−11	1.25E−05	2.68E−06	NA
C25	3.02E−11	7.61E−06	2.26E−11	3.02E−11	3.02E−11	4.22E−04	NA
C26	1.95E−11	NA	1.24E−11	1.95E−11	1.95E−11	6.27E−08	1.04E−08
C27	7.39E−11	6.61E − 02	2.02E−05	3.02E−11	8.89E−10	7.22E−06	NA
C28	3.02E−11	1.94E−10	2.11E−11	3.02E−11	3.02E−11	9.92E−11	NA
C29	2.50E−11	NA	1.61E−11	2.50E−11	2.50E−11	4.14E−11	9.74E−05
C30	3.02E−11	1.21E−04	1.95E−11	3.02E−11	3.02E−11	1.96E−10	NA

results showed the improved efficacy of the proposed BMWOA in dealing with various problems.

Future studies can focus on the efficacy of the BMWOA in various applications that require enhanced local search trends. Another opportunity is to further enhance the efficiency of the BMWOA in terms of exploration and exploitation and apply it to another set of benchmark cases.

Compliance with ethical standards

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

Conflict of interest There is no conflict of interest to declare.

References

1. Abbassi R, Abbassi A, Heidari AA, Mirjalili S (2019) An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. Energy Convers Manag 179:362–372
2. Abdel-Basset M, Abdle-Fatah L, Sangaiah AK (2018) An improved lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment. Cluster Comput. https://doi.org/10.1007/s10586-018-1769-z
3. Abdel-Basset M, El-Shahat D, El-Henawy I, Sangaiah AK (2018) A modified flower pollination algorithm for the multidimensional knapsack problem: human-centric decision making. Soft Comput 22(13):4221–4239
4. Abdel-Basset M, El-Shahat D, El-henawy I, Sangaiah AK, Ahmed SH (2018) A novel whale optimization algorithm for cryptanalysis in Merkle-Hellman cryptosystem. Mobile Netw Appl 23(4):1–11
5. Abdel-Basset M, Hessin AN, Abdel-Fatah L (2018) A comprehensive study of cuckoo-inspired algorithms. Neural Comput Appl 29(2):345–361

6. Abdel-Basset M, Manogaran G, Abdel-Fatah L, Mirjalili S (2018) An improved nature inspired meta-heuristic algorithm for 1-d bin packing problems. Pers Ubiquitous Comput 22(5):1–16
7. Abdel-Basset M, Manogaran G, El-Shahat D, Mirjalili S (2018) A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem. Future Gener Comput Syst 85:129–145
8. Abdel-Basset M, Manogaran G, El-Shahat D, Mirjalili S (2018) Integrating the whale algorithm with tabu search for quadratic assignment problem: a new approach for locating hospital departments. Appl Soft Comput 73:530–546
9. Al-Betar MA (2016) Beta-hill climbing: an exploratory local search. Neural Comput Appl 28(1):1–16
10. Aljarah I, Faris H, Mirjalili S (2016) Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Comput 22(1):1–15
11. Aljarah I, Mafarja M, Heidari AA, Faris H, Zhang Y, Mirjalili S (2018) Asynchronous accelerating multi-leader salp chains for feature selection. Appl Soft Comput 71:964–979
12. Awad NH, Ali MZ, Suganthan PN, Reynolds RG (2017) Cade: a hybridization of cultural algorithm and differential evolution for numerical optimization. Inf Sci 378:215–241
13. Chen J, Xin B, Peng Z, Dou L, Zhang J (2009) Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization. IEEE Trans Syst Man Cybern Part A Syst Hum 39(3):680–691
14. Chen J, Zheng J, Wu P, Zhang L, Wu Q (2017) Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems. Expert Syst Appl 86:208–223
15. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut Comput 1(1):3–18
16. El-Abd M, Kamel M (2005) A taxonomy of cooperative search algorithms. Hybrid Metaheuristics 3636:32–41
17. El Aziz MA, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256
18. Eriksen N, Miller LA, Tougaard J, Helweg DA (2005) Cultural change in the songs of humpback whales (megaptera novaeangliae) from tonga. Behaviour 142(3):305–328
19. Faris H, Al-Zoubi AM, Heidari AA, Aljarah I, Mafarja M, Hassonah MA, Fujita H (2019) An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. Inf Fusion 48:67–83. https://doi.org/10.1016/j.inffus.2018.08.002
20. Faris H, Mafarja MM, Heidari AA, Aljarah I, Ala'M AZ, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowl Based Syst 154:43–67
21. Gandomi AH, Yang XS, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89(23):2325–2336
22. Gao Y, Du W, Yan G (2015) Selectively-informed particle swarm optimization. Sci Rep 5:9295
23. Greiner R (1996) Palo: a probabilistic hill-climbing algorithm. Artif Intell 84(1–2):177–208
24. Heidari AA, Abbaspour RA, Jordehi AR (2017) An efficient chaotic water cycle algorithm for optimization tasks. Neural Comput Appl 28(1):57–85
25. Heidari AA, Abbaspour RA, Jordehi AR (2017) Gaussian barebones water cycle algorithm for optimal reactive power dispatch in electrical power systems. Appl Soft Comput 57:657–671
26. Heidari AA, Faris H, Aljarah I, Mirjalili S (2018) An efficient hybrid multilayer perceptron neural network with grasshopper optimization. Soft Comput. https://doi.org/10.1007/s00500-018-3424-2
27. Heidari AA, Pahlavani P (2017) An efficient modified grey wolf optimizer with lévy flight for optimization tasks. Appl Soft Comput 60:115–134
28. Jordehi AR (2015) Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems. Appl Soft Comput 26:401–417
29. Jordehi AR (2015) A review on constraint handling strategies in particle swarm optimisation. Neural Comput Appl 26(6):1265–1275
30. Jordehi AR (2016) Time varying acceleration coefficients particle swarm optimisation (TVACPSO): a new optimisation algorithm for estimating parameters of PV cells and modules. Energy Convers Manag 129:262–274
31. Jordehi AR (2018) Enhanced leader particle swarm optimisation (ELPSO): an efficient algorithm for parameter estimation of photovoltaic (PV) cells and modules. Solar Energy 159:78–87
32. LaTorre A, Peña JM (2017) A comparison of three large-scale global optimizers on the CEC 2017 single objective real parameter numerical optimization benchmark. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE, pp 1063–1070
33. Li R, Hu S, Wang Y, Yin M (2017) A local search algorithm with tabu strategy and perturbation mechanism for generalized vertex cover problem. Neural Comput Appl 28(7):1775–1785
34. Lin SW, Lee ZJ, Ying KC, Lee CY (2009) Applying hybrid metaheuristics for capacitated vehicle routing problem. Expert Syst Appl 36(2):1505–1512
35. Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier-Viger P, Li X, Mirjalili S (2018) Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowl Based Syst 161:185–204. https://doi.org/10.1016/j.knosys.2018.08.003
36. Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Ala'M AZ, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl Based Syst 145:25–45. https://doi.org/10.1016/j.knosys.2017.12.037
37. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260:302–312
38. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133
39. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
40. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
41. Mohamed AAA, Mohamed YS, El-Gaafary AA, Hemeida AM (2017) Optimal power flow using moth swarm algorithm. Electr Power Syst Res 142:190–206
42. Oliva D, El Aziz MA, Hassanien AE (2017) Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm. Appl Energy 200:141–154
43. Parks SE, Cusano DA, Stimpert AK, Weinrich MT, Friedlaender AS, Wiley DN (2014) Evidence for acoustic communication among bottom foraging humpback whales. Sci Rep 4:7508
44. Ramp C, Hagen W, Palsbøll P, Bérubé M, Sears R (2010) Age-related multi-year associations in female humpback whales (megaptera novaeangliae). Behav Ecol Sociobiol 64(10):1563–1576
45. Reddy VV, Manohar TG et al (2017) Optimal renewable resources placement in distribution networks by combined power loss index and whale optimization algorithms. J Electr Syst Inf Technol 28:669–678
46. Rendell L, Whitehead H (2001) Culture in whales and dolphins. Behav Brain Sci 24(2):309–324

47. Simon D (2008) Biogeography-based optimization. IEEE Trans Evolut Comput 12(6):702–713
48. Tharwat A, Moemen YS, Hassanien AE (2017) Classification of toxicity effects of biotransformed hepatic drugs using whale optimized support vector machines. J Biomed Inform 68:132–149
49. Tsamardinos I, Brown LE, Aliferis CF (2006) The max–min hill-climbing bayesian network structure learning algorithm. Mach Learn 65(1):31–78
50. Wang L, Zeng Y, Chen T (2015) Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Syst Appl 42(2):855–863
51. Yang XS, Hossein Gandomi A (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483
52. Yang XS, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multiobjective optimization. Eng Optim 46(9):1222–1237