Improved Whale Optimization Algorithm Based on Inertia Weights for Solving Global Optimization Problems

I-Ming Chao[1], Shou-Cheng Hsiung[1,*], Jenn-Long Liu[2]

[1]Department of Industrial Management, I-Shou University, Kaohsiung, Taiwan

[2]Department of Information Management, I-Shou University, Kaohsiung, Taiwan

Received 08 September 2019; received in revised form 09 December 2019; accepted 20 February 2020

DOI: https://doi.org/10.46604/aiti.2020.4167

Abstract

Whale Optimization Algorithm (WOA) is a new kind of swarm-based optimization algorithm that mimics the foraging behavior of humpback whales. WOA models the particular hunting behavior with three stages: encircling prey, bubble-net attacking, and search for prey. In this work, we proposed a new linear decreasing inertia weight with a random exploration ability (LDIWR) strategy. It also compared with the other three inertia weight WOA (IWWOA) methods: constant inertia weight (CIW), linear decreasing inertia weight (LDIW), and linear increasing inertia weight (LIIW) by adding fixed or linear inertia weights to the position vector of the reference whale. The four IWWOAs are tested with 23 mathematical and theoretical optimization benchmark functions. Experimental results show that most of IWWOAs outperform the original WOA in terms of solution accuracy and convergence rate when solving global optimization problems. Accordingly, the LDIWR strategy produces a better balance between exploration and exploitation capabilities for multimodal functions.

Keywords: whale optimization algorithm, bubble-net feeding method, inertia weights, exploration and exploitation capabilities

1. Introduction

Optimization plays essential roles in scientific research, management, and industry because numerous real-world problems can mostly model as optimization tasks [1]. In recent years, many meta-heuristic algorithms have been widely applied as powerful tools to solve optimization problems. Due to the following reasons: (i) have fewer parameters; (ii) do not require gradient information; (iii) can bypass local optima; (iv) can be utilized to solve the practical problems [2].

There are many popular population-based meta-heuristic optimization algorithms, such as Particle Swarm Optimization (PSO) [3], Ant Colony System (ACS) [4], Artificial Bee Colony (ABC) [5], Cuckoo Search (CS) [6], Fruit fly Optimization Algorithm [7], and Whale Optimization Algorithm (WOA) [8]. Among them, WOA, proposed by Mirjalili and Lewis [8], is competitive. Due to the simplicity of WOA in implementation and only two main parameters adjusted, the algorithm has shown superior compared to the state-of-art meta-heuristic algorithms from the testing results of different benchmark functions and engineering design problems [8-9]. Basically, the algorithm is inspired by the foraging behavior of humpback whales.

Humpback whales in a group hunt school of krill or small fishes by shrinking and encircling around them to herd them to close to the sea surface and generating bubbles along a helix-shaped or '9'-shaped path to perform a bubble-net attack [8-9]. Population-based meta-heuristic optimization algorithms have a common feature regardless of their nature — the search process dividing into two phases: exploration and exploitation [8, 10]. The mechanisms of shrinking encircling and spiral

[*] Corresponding author. E-mail address: melvinisstrong@gmail.com

updating represent the exploitation phase, and the method of a random search for prey represents the exploration phase. Finding a proper balance between exploration and exploitation is the most challenging task in the development of any metaheuristic algorithm due to the stochastic nature of the optimization process.

The main problems faced by WOA are slow and premature convergence, similar to other meta-heuristic algorithms. Therefore, many variants of WOA were proposed in the literature. To enhance the convergence speed and exploitation mechanism, Mafarja and Mirjalili proposed a memetic algorithm by hybrid WOA with Simulated Annealing (SA) by searching the most promising regions located by the WOA algorithm to improve the exploitation [9]. Because the updated solution is mainly depending on the current best solution got so far. Hu et al. [2] proposed the original WOA with inertia weights, which is similar to the modified PSO algorithm [11], to get an improved WOA. In 2018, Kaur and Arora proposed a Chaotic Whale Optimization Algorithm (CWOA) to replace the critical parameter ‘p’ of WOA instead of 0.5 probability that whale either chose the encircling or spiral path to update the position during optimization in the original WOA [12].

To conclude the literature review; different inertia weight strategies may get various incremental changes in a better solution. In this paper, a new idea was proposed by adding linear decreasing inertia weight to the position vector of the reference whale and remaining the exploration ability of the original WOA. To compare with the two best inertia weight methods, obtained from the study of Hu et al. [2], the efficiency of the opposite strategy is also observed for linear decreasing inertia weight method. The rest of this paper is organized as follows. Section 2 presents the basis of the original WOA algorithm. Improving WOA by four inertia weightings was performed in Section 3. In Section 4, the experimental results presented and result analyzed. Finally, in Section 5, conclusions are given.

2. Whale Optimization Algorithm

WOA is inspired by the individual foraging behavior of humpback whales. Fig. 1 [9] shows the special hunting method of humpback whales. WOA models the behavior as two phases. The first one is the exploitation phase, including encircling prey and spiral bubble-net attacking method. The second one is the exploration phase when searching randomly for prey.

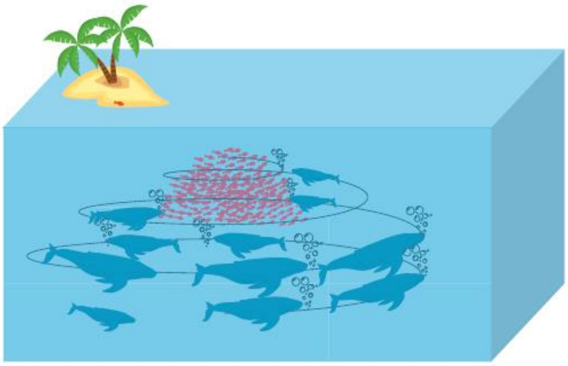


Fig. 1 Bubble-net feeding behavior of humpback whales [9]

2.1. Exploitation phase (encircling prey and spiral bubble-net attacking method)

Humpback whales hunt prey with two steps: encircling around them and creating bubbles-nets. First, they recognize the locations of the prey and then encircle around them. WOA algorithm assumes that the solution of the current best candidate (leader whale) is targeting prey or closing to the optimum. When encircling the prey, the other whales update their positions towards the best whale obtained so far. Therefore, encircling prey can be represented by the following equations [8]:

$$X_i^j(t+1) = X_p^k(t) - A \cdot D_j \tag{1}$$

$$D_j = \left| C \cdot X_p^j(t) - X_i^j(t) \right| \tag{2}$$

The number of population and variables (dimensions) is assumed as M, and N. X_i^j is a position matrix, $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$ represents the position of the *i*th whale on the *j*th dimension. X_p^j represents the position of prey (the leader whale P) on the *j*th dimension. Where t indicates the current iteration, $X_p^j(t)$ represents the current position on the *j*th dimension of the leader whale P, and $X_i^j(t)$ represents the current position of the *i*th whale on *j*th dimension; | |denotes the absolute value. X_p should be updated every iteration if there is a better solution appearing. *A* and *C* are coefficient numbers that are calculated based on random functions as follows:

$$A = 2a \cdot rand_1 - a \tag{3}$$

$$C = 2a \cdot rand_2 \tag{4}$$

$$a = 2 - t\frac{2}{MaxIter} \tag{5}$$

Where MaxIter is the maximum number of iterations, α is linearly decreasing from 2 to 0 from the start to the end of iterations (in both exploration and exploitation phases). Since $rand_1$ and $rand_2$ are random numbers in the intervals [0, 1], then the ranges of *A* and *C* are in the intervals $[-\alpha, \alpha]$ and [0, 2], respectively.

In the exploitation phase, whales swim around the prey within the shrinking circle as well as move along a spiral-shaped path at the same time to form distinctive bubbles along a 9-shaped path to perform the bubble-net attacking [8]. In other words, there are two types of behavior in the bubble-net attacking that include a shrinking encircling mechanism or spiral updating position. As mentioned before, the value of a is linearly decreased from 2 to 0. Since the range of *A* being in the intervals$[-\alpha, \alpha]$, the absolute value of *A* decreases for iterations. Therefore, the shrinking encircling mechanism can be achieved by using Eq. (1) The spiral-shaped path between a whale $X_i^j(t)$ and the prey $X_p^j(t)$ can be expressed by Eq. (6) [9].

$$X_i^j(t+1) = D'_j \cdot e^{bl}\cos(2\pi l) + X_p^j(t) \tag{6}$$

Where $D'_j = |X_p^j(t) - X_i^j(t)|$ represents the distance between the *i*th whale and the prey (the best solution obtained so far) on the *j*th dimension, b is a constant for defining the logarithmic spiral shape, and *l* is a random number in [-1, 1]. To model the two mechanisms in the bubble-net attacking, shrinking encircling, and the spiral-shaped path. Mirjalili and Lewis assumed that there is a probability of 0.5 to choose between them throughout iterations as in Eq. (7) [8, 9].

$$X_i^j(t+1) = \begin{cases} X_p^j(t) - A \cdot D_j & if\ p \prec 0.5 \\ D'_j \cdot e^{bl}\cos(2\pi l) + X_p^j(t) & if\ p \geq 0.5 \end{cases} \tag{7}$$

2.2. *Exploration phase* (*search for prey*)

To enhance the exploration mechanism in WOA, instead of updating the positions by the location of the best solution found so far. A random whale is selected to guide the search. When $|A| \succ 1$, a whale forced to move far away from the best-known whale to execute the global search for prey (exploration). However, when $|A| \prec 1$, a whale will perform a local search according to the best solution found so far (exploitation). This mechanism can model as follows [8, 9]:

$$D_j = \left|C \cdot X_{rand}^j(t) - X_i^j(t)\right| \tag{8}$$

$$X_i^j(t+1) = X_{rand}^j(t) - A \cdot D_j \tag{9}$$

where $X_{rand}^j(t)$ represents the position on *j*th dimension of a random whale from the population selected by $|A| \succ 1$.

3. Inertia Weight WOA

As seen in the previous section 2, the original WOA updated solution mostly depends on the current best candidate solution. From the literature review, different inertia weight strategies are known to get different changes in solution [2, 11]. Inertia weights are introduced ω & ωr ∈ [0, 1] into WOA to get the improved whale optimization algorithm (Inertia Weight Whale Optimization Algorithm, IWWOA) by adding fixed or linear inertia weights to the position of the reference whale.

In the exploitation phase ($|A| \prec 1$), the updated method is represented by the following equations:

$$D_j = \left|C \cdot \omega \cdot X_p^j(t) - X_i^j(t)\right| \tag{10}$$

$$X_i^j(t+1) = \begin{cases} \omega \cdot X_p^j(t) - A \cdot D_j & if\ p \prec 0.5 \\ D_j' \cdot e^{bl} \cos(2\pi l) + \omega \cdot X_p^j(t) & if\ p \geq 0.5 \end{cases} \tag{11}$$

In the exploration phase ($|A| \succ 1$), the updated mathematical model is as follows:

$$D_j = \left|\omega r \cdot C \cdot X_{rand}^j(t) - X_i^j(t)\right| \tag{12}$$

$$X_i^j(t+1) = \omega r \cdot X_{rand}^j(t) - A \cdot D_j \tag{13}$$

Hu et al. [2] had proved to add inertia weights into WOA are competitive with other meta-heuristic algorithms: WOA, FOA, ABC, and PSO can enhance the ability of exploitation. Hu et al. [2] applied four strategies, the LDIW strategy introduced by Shi and Eberhart [11] in 1998, Sugeno function as an inertia weight (SFIW) introduced by Lei et al. [13] in 2006, exponential decreasing inertia weight (EDIW) strategy and CIW introduced by Lu et al. [14] in 2014. IWWOA1 is the best result of the study [2] by adding smaller fixed inertia weights to the position vector of the reference whale. IWWOA2 uses the better region refer to the paper [11] by adding linear inertia weights to the reference whale to strengthen the local search ability of the original WOA. IWWOA3 is the opposite way of IWWOA2. The IWWOA4 represents the new idea by adding linear decreasing inertia weight to the position vector of the reference whale and remaining the exploration ability of the original WOA in the meantime. Those ways are in the following:

(1) IWWOA1: ω = ωr = 0.1 (CIW-the best);

(2) IWWOA2: ω = ωr; ω_initial=0.9; ω_final=0.4; ω=ω_initial-(t/Max_iter)×(ω_initial-ω_final) (LDIW-revised);

(3) IWWOA3: ω = ωr; ω_initial=0.4; ω_final=0.9; ω=ω_initial-(t/Max_iter)×(ω_initial-ω_final) (LIIW-revised);

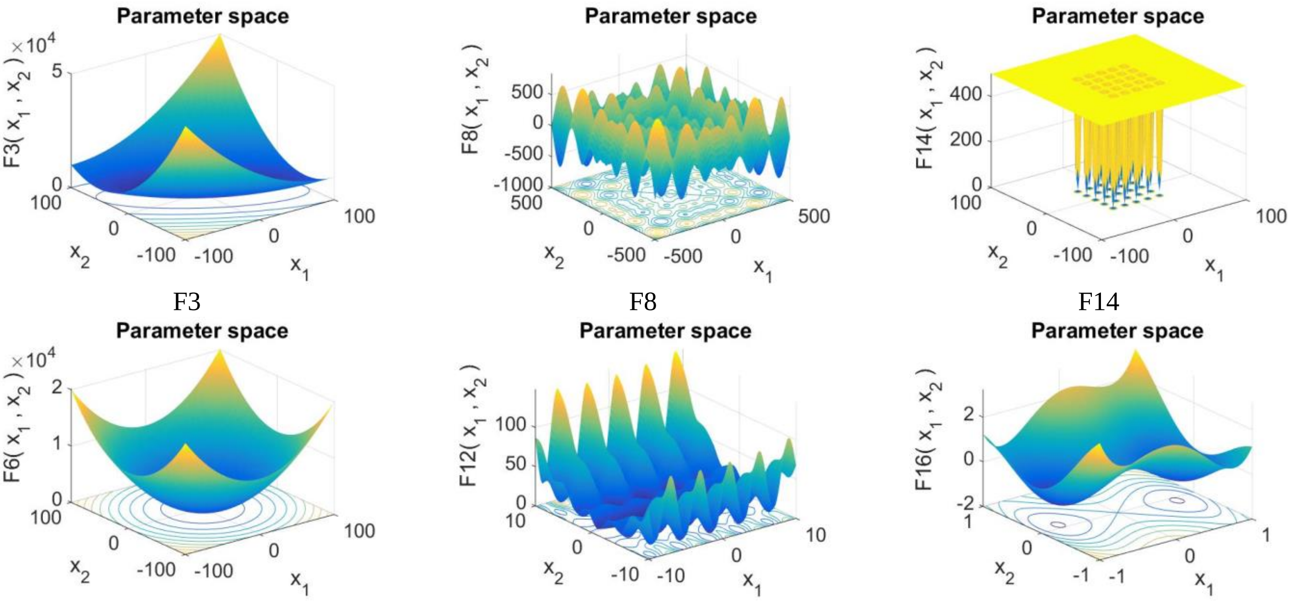
(4) IWWOA4: ωr=1; ω_initial=0.9; ω_final=0.4; ω=ω_initial-(t/Max_iter)×(ω_initial-ω_final) (LDIWR-proposed)

where ω and ωr represent the inertia weight in the exploitation and exploration phase, respectively. When ω = ωr = 1, IWWOAs become the original WOA. ω = ωr = 0.1 have already been known can obtain a better solution in the CIW strategy for WOA [2]. Inertia weight linearly was learned to decrease from ω_initial=0.9 to ω_final=0.4 can obtain a better solution in the LDIW strategy for PSO [11]. The effect of the LIIW strategy on WOA wants to be realized (from ω_initial=0.4 to ω_final=0.9), and the effect of the LDIWR strategy on WOA.

4. Results and Discussion

4.1. Experiments of benchmark functions

To test the performance of the original WOA [8] and the LDIWR strategy, the former methods proposed by Hu et al. [2], the same 23 benchmark functions used in the literature [8] are taken, which consist of 7 (F1-F7) unimodal, 6 (F8-F13) multimodal and 10 (F14-F23) fixed- dimension multimodal functions. Those functions are quite different from the test functions of Hu et al. [2] in 2017 when consisting of 18 unimodal functions and only 9 multimodal functions. Fig. 2 shows the typical 2D plot examples of the test benchmark functions considered in this work.



F3 F8 F14

F6 F12 F16

(a) Unimodal (b) Multimodal (c) Fixed- dimension multimodal

Fig. 2 Typical 2D plots of benchmark functions

For all IWWOAs & WOA algorithms, the population size and maximum iteration are equal to 30 and 500, respectively. For each benchmark function, IWWOAs & WOA algorithms were executed independently run 30 times. Table 1 and Table 2 compares the best, the best mean (mean) and standard deviation (STD) of solutions obtained by using the original WOA & IWWOAs for the benchmark functions (F1-F23).

4.2. *Evaluation of exploitation capability* (*functions F1–F7*)

F1-F7 are unimodal functions because there is only one global optimum solution. These functions usually utilized to evaluate the exploitation ability of meta-heuristic algorithms [8]. As seen from Table 1 and Table 2, the smaller CIW strategy got 5 in 7 of the best means, and the LDIW and LDIWR strategy got the others. The IWWOAs are highly competitive with the original WOA algorithm. Most of the IWWOAs is better than the original WOA on the mean and standard deviation of 30 independent runs. Furthermore, comparing the LDIW and LIIW methods, the performance of those approaches follows this order: LDIW > LIIW. Because the LDIW strategy (ω = ωr from 0.9 to 0.4) leads the smaller and smaller stride, so it can provide better exploitation ability than LIIW (ω = ωr from 0.4 to 0.9), especially in the final stage to find the globally optimal solution of unimodal functions. It is worth mentioning that the IWWOA1 (CIW ω = ωr = 0.1) is the best method on F1, F2, F3, F4, and F7. Thus, IWWOAs provide excellent exploitation capability, and the smaller CIW strategy is the better choice for unimodal functions.

Table 1 Comparison of the best, mean and std values of the objective functions obtained using the WOA and IWWOA1-IWWOA2 (continued)

Fun	Original WOA		IWWOA1(0.1) CIW		IWWOA2(0.9->0.4) LDIW	
F1	**1.60450e-73**	**6.09550e-73**	**0.00000e+00**	**0.00000e+00**	1.54950e-271	**0.00000e+00**
F2	**3.90430e-49**	**2.01600e-48**	**4.50300e-220**	**0.00000e+00**	6.22180e-144	1.63220e-143
F3	**4.46769e+04**	**1.42133e+04**	**0.00000e+00**	**0.00000e+00**	7.35480e-182	**0.00000e+00**
F4	**5.46388e+01**	**2.87173e+01**	**1.43180e-218**	**0.00000e+00**	3.12300e-120	1.61210e-119

Table 1 Comparison of the best, mean and std values of the objective functions obtained using the WOA and IWWOA1-IWWOA2

Fun	Original WOA		IWWOA1(0.1) CIW		IWWOA2(0.9->0.4) LDIW	
F5	2.79790E+01	**3.73070E-01**	**2.87286E+01**	**3.23700E-02**	2.79719E+01	2.92100E-01
F6	4.19590E-01	2.67260E-01	**6.09030E-01**	**3.07770E-01**	**2.81010E-01**	1.25790E-01
F7	**4.48670E-03**	**5.20030E-03**	**6.78090E-05**	**6.12410E-05**	8.85760E-05	7.35660E-05
sum	**5 / 0**	**6 / 0**	**2 / 5**	**1 / 6**	**0 / 1**	**0 / 2**
F8	-9.95404E+03	1.56063E+03	**-7.60267E+03**	**2.17479E+03**	-1.10905E+04	1.63330E+03
F9	**1.89480E-15**	**1.02040E-14**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
F10	**4.79620E-15**	**2.31150E-15**	**8.88180E-16**	**0.00000E+00**	2.42770E-15	1.76050E-15
F11	**6.73470E-03**	**3.62670E-02**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
F12	**4.06280E-02**	**7.38760E-02**	3.70340E-02	2.43910E-02	1.24130E-02	5.75200E-03
F13	**5.60110E-01**	2.38900E-01	4.01050E-01	**2.49290E-01**	2.35320E-01	8.37050E-02
F14	3.12180E+00	3.50270E+00	**6.09460E+00**	**3.61850E+00**	2.60110E+00	3.01320E+00
F15	6.61020E-04	3.82440E-04	**8.33010E-04**	**6.12800E-04**	7.54130E-04	5.87520E-04
F16	**-1.03160E+00**	**1.32260E-09**	**-9.83610E-01**	**1.86660E-02**	-1.02900E+00	2.95320E-03
F17	**3.97900E-01**	**1.16820E-05**	**5.99420E-01**	**1.64920E-01**	3.98180E-01	4.20450E-04
F18	**3.00010E+00**	**1.57690E-04**	**1.02615E+01**	**9.21590E+00**	3.01600E+00	2.36150E-02
F19	**-3.85460E+00**	**1.00220E-02**	**-3.26660E+00**	**4.38600E-01**	-3.83800E+00	2.26320E-02
F20	-3.21410E+00	1.13040E-01	**-1.80080E+00**	**4.58620E-01**	**-3.24200E+00**	**9.04730E-02**
F21	**-8.17300E+00**	2.58430E+00	**-2.65770E+00**	1.11060E+00	-7.31410E+00	**2.77040E+00**
F22	-7.58410E+00	**3.06820E+00**	**-2.56330E+00**	**9.32650E-01**	-7.09060E+00	2.74490E+00
F23	-6.83310E+00	**3.32870E+00**	**-2.90990E+00**	1.65360E+00	-6.44140E+00	2.50680E+00
sum	**5 / 5**	**6 / 4**	**11 / 3**	**9 / 3**	**0 / 3**	**1 / 3**

Remarks:
1. Data with bold black font indicates the worst among the original WOA and the other IWWOAs
2. Data with bold red font indicates the best among the original WOA and the other IWWOAs
3. The rows of sum indicate the accumulated number of the worst and the best in unimodal and multimodal functions

4.3. *Evaluation of exploration capability* (*functions F8–F23*)

Multimodal functions are more complex than unimodal functions. The former functions include many local optima whose complexity increases exponentially with the number of design variables. Consequently, this kind of test problems turns very popular to evaluate the exploration ability of the optimizer [8]. The results reported in Table 1 and Table 2 for functions F8-F23 show that the LDIWR strategy has an outstanding exploration capability. Comparing the performance of the original WOA with the inertia weighting methods in this work follows the order: LDIWR > original WOA > LIIW > LDIW > smaller CIW. The novel LDIWR method obtained 7 best mean solutions on F8-F9, F11-F14, and F23 in the 16 multimodal functions. Notably, there is no worst solution obtained 8 best mean solutions on F8-F9, F11-F14, and F22-F23 in the 16 multimodal functions. Notably, there is no worst solution obtained by the LDIWR strategy. Previously mentioned LDIWR integrated the exploration mechanism of the original WOA algorithm (ωr=1), and the better exploitation mechanism of the LDIW strategy on WOA, as mentioned before. However, there are 11 worst mean solutions obtained by the CIW approach. It might be decreasing the exploration capability of IWWOA1 by using the smaller CIW strategy (ω = ωr = 0.1). It is worth mentioning that the LIIW strategy is better than the LDIW for multimodal functions. Because the LIIW (ω = ωr from 0.4 to 0.9) leads the larger and larger stride; it could provide better exploration ability than the LDIW (ω = ωr from 0.9 to 0.4), especially in the final stage to find the globally optimal solution of multimodal functions. Hence, the LDIWR is more robust than the other IWWOAs and the original WOA.

Table 2 Comparison of the best, mean and std values of the objective functions obtained using the IWWOA3 and IWWOA4

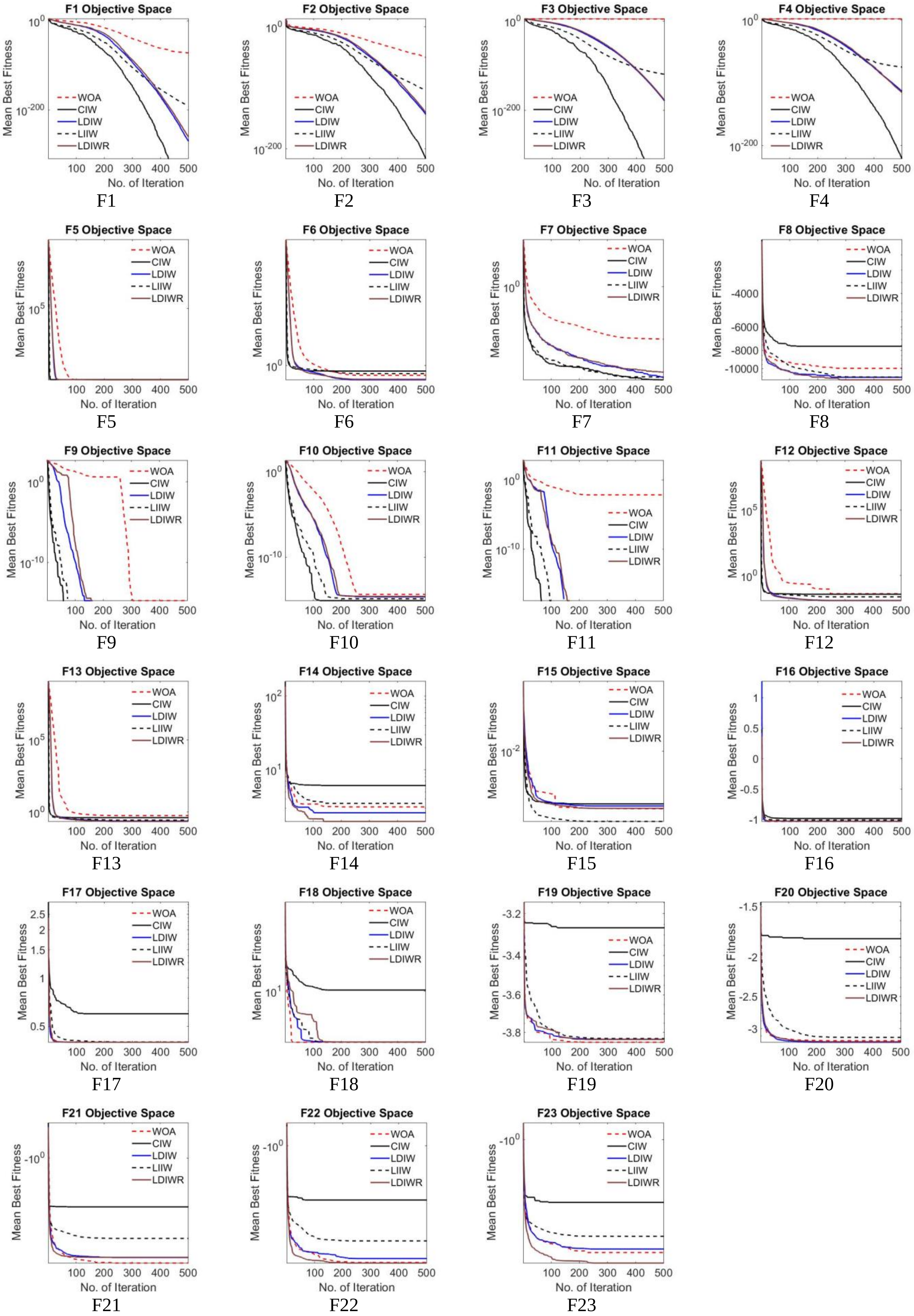
Fun	IWWOA3(0.4->0.9) LIIW		IWWOA4(0.9->0.4) with Random ability-LDIWR	
	mean	std	mean	std
F1	1.19800E-190	**0.00000E+00**	1.86600E-261	**0.00000E+00**
F2	4.25480E-105	1.38300E-104	1.71000E-141	5.32650E-141
F3	6.60840E-120	2.78580E-119	6.52640E-178	**0.00000E+00**
F4	1.09580E-75	3.82330E-75	2.99650E-116	1.56740E-115
F5	2.81399E+01	2.80560E-01	**2.78947E+01**	2.43210E-01
F6	4.88890E-01	1.75960E-01	2.95130E-01	**9.86380E-02**
F7	8.46060E-05	8.72020E-05	1.49770E-04	1.20400E-04
sum	**0 / 0**	**0 / 1**	**0 / 1**	**0 / 3**
F8	-1.10724E+04	1.68468E+03	**-1.14430E+04**	**1.51963E+03**
F9	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
F10	1.48030E-15	1.32400E-15	2.78300E-15	1.77240E-15
F11	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
F12	2.33280E-02	8.77710E-03	**1.20190E-02**	**4.66670E-03**
F13	2.70880E-01	8.10020E-02	**2.19660E-01**	**7.98360E-02**
F14	3.48570E+00	3.34880E+00	**1.98310E+00**	**2.10240E+00**
F15	**3.63690E-04**	**8.53780E-05**	6.74530E-04	5.41630E-04
F16	-1.00900E+00	1.12530E-02	-1.02780E+00	5.95660E-03
F17	3.99250E-01	1.95610E-03	3.98160E-01	3.54870E-04
F18	3.01550E+00	2.42980E-02	3.01610E+00	2.44000E-02
F19	-3.83410E+00	2.84740E-02	-3.83910E+00	2.49720E-02
F20	-3.15340E+00	1.28960E-01	-3.23220E+00	9.90000E-02
F21	-4.99030E+00	**6.67050E-02**	-7.29060E+00	2.40980E+00
F22	-5.22030E+00	**8.95650E-01**	**-7.66510E+00**	2.64640E+00
F23	-5.17970E+00	**9.96500E-01**	**-8.17640E+00**	2.53490E+00
sum	**0 / 3**	**0 / 6**	**0 / 8**	**0 / 6**

Remarks:
1. Data with bold black font indicates the worst among the original WOA and the other IWWOAs
2. Data with bold red font indicates the best among the original WOA and the other IWWOAs
3. The rows of sum indicate the accumulated number of the worst and the best in unimodal and multimodal

4.4. Analysis of convergence behavior

As learned that the quality of the solution and the convergence speed of an algorithm of the global optimal solution enormously depends on the parameter and search strategy of the optimizer [1]. Therefore; analyzing the convergence behavior after different inertia weighting was added to the position vector of the reference whale in necessary. Convergence curves of the original WOA, CIW, LDIW, LIIW, and LDIWR were compared in Fig. 3 for 23 benchmark functions. IWWOAs are competitive enough with the original WOA on the mean best fitness in each iteration over 30 runs. The slow convergence and premature convergence problems of WOA on unimodal and multimodal functions (F1 to F13) can be seen.

As shown in Fig. 3, the IWWOAs shows three different convergence behaviors when optimizing 23 benchmark functions. Firstly, the convergence of the IWWOAs tends to be accelerated as iteration increases except the LIIW on F1 through F4. Secondly, IWWOAs trend of convergence within fewer iterations. The adding inertia weight strategy proposed for CIW, LDIW, and LIIW restrict the exploration capability to assist them in looking for the promising regions of unimodal and multimodal functions in the initial steps of iteration. They also cause a more rapid converge towards the optimum almost before half of the iterations. This behavior is evident in F5 through F13. Thirdly, the original WOA and LDIWR are almost faster than the others on the fixed-dimension multimodal functions. Their excellent performance is due to their full exploration ability of WOA on multimodal functions.



F1 F2 F3 F4

F5 F6 F7 F8

F9 F10 F11 F12

F13 F14 F15 F16

F17 F18 F19 F20

F21 F22 F23

Fig. 3 Comparison of convergence curves obtained using the original WOA and proposed four IWWOAs for solving 23 benchmark functions

5. Conclusions

This work introduced four inertia weighting to the position vector of the reference whale to strengthen the local search ability of the WOA. The proposed IWWOAs on 23 mathematics benchmark functions were conducted to analyze the exploitation, exploration, and convergence behavior by comparison with the original WOA, IWWOA1 (CIW ω = ωr = 0.1), IWWOA2 (LDIW ω = ωr = 0.9 to 0.4), IWWOA3 (LIIW ω = ωr = 0.4 to 0.9), and IWWOA4 (LDIWR ω =0.9 to 0.4, ωr =1).

IWWOAs were found to be competitive enough. According to our analysis, the smaller constant inertia weight strategy (CIW ω = ωr = 0.1) was the better choice for unimodal functions. Furthermore, the linear decreasing inertia weight with random exploration ability strategy (LDIWR ω =0.9 to 0.4, ωr =1) preserved the full exploration capability of WOA. It possesses the benefit of the inertia weight method, which is also more robust than the other IWWOAs and the original WOA on multimodal functions. Hence, the LDIWR strategy can produce a better balance between exploration and exploitation capabilities for searching solutions and result in an improvement in the convergence speed and optimal solution of the original WOA.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] G. Wu, R. Mallipeddi, and P. N. Suganthan, “Ensemble strategies for population-based optimization algorithms- A survey,” Swarm and Evolutionary Computation, vol. 44, pp. 695-711, February 2019.

[2] H. Hu, Y. Bai, and T. Xu, “Improved whale optimization algorithms based on inertia weights and theirs applications,” International Journal of Circuits, Systems and Signal Processing, vol. 11, pp. 12-26, 2017.

[3] J. Kenney and R. Eberhart “Particle swarm optimization,” Proc. IEEE Symp. International Conference on Neural Networks, IEEE Press, 1995, pp. 1942-1948.

[4] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53-66, April 1997.

[5] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm,” Applied Soft Computing, vol. 8, no 1, pp. 687-697, February 2008.

[6] X. S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” International Journal of Mathematical Modelling and Numerical Optimisation, vol. 1, no. 4, pp. 330-343, December 2010.

[7] W. T. Pan, “A new fruit fly optimization algorithm: taking the financial distress model as an example,” Knowledge-Based Systems, vol. 26, pp. 69-74, February 2012.

[8] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” Advances in Engineering Software, vol. 95, pp. 51-67, May 2016.

[9] M. M. Mafarja and S. Mirjalili, “Hybrid whale optimization algorithm with simulated annealing for feature selection,” Neurocomputing, vol. 260, pp. 302-312, October 2017.

[10] M. Črepinšek, S. H. Liu, and M. Mernik, “Exploration and exploitation in evolutionary algorithms: a survey,” ACM Computing Surveys (CSUR), vol. 45, no. 3, pp. 1-33, June 2013.

[11] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization,” Proc. International Conference on Evolutionary Programming, March 1998, pp. 591-600.

[12] G. Kaur and S. Arora, “Chaotic whale optimization algorithm,” Journal of Computational Design and Engineering, vol. 5, no. 3, pp. 275-284, January 2018.

[13] K. Lei, Y. Qiu, and Y. He, “A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization,” Proc. IEEE Symp. 2006 1st International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA), January 2006, pp. 977-980.

[14] J. Lu, H. Hu, and Y. Bai, “Radial basis function neural network based on an improved exponential decreasing inertia weight-particle swarm optimization algorithm for AQI prediction,” Abstract and Applied Analysis, vol. 2014, pp. 1-9, July 2014.