An adaptive optimisation algorithm based on modified whale optimisation algorithm and Laplace crossover

Lamiaa M. El Bakrawy

Faculty of Science,
Al-Azhar University,
Cairo, Egypt
Email: dr_lamiaa_el_bakrawy@azhar.edu.eg

Abstract: Whale optimisation algorithm (WOA) is a new bio-inspired algorithm which mimics the hunting behaviour of humpback whale in nature. Standard WOA is easily trapped in local optima, provide slow convergence rate and lack of diversity, as the dimension of the search space expansion. In this paper, modified whale optimisation algorithm (MWOA) is proposed to improve the quality of standard WOA algorithm performance. Moreover, an adaptive optimisation algorithm based on modified whale optimisation algorithm and Laplace crossover (ALMWOA) is presented in this paper to increase the diversity of search space and enhance the capability to avoid local optimal solutions. The proposed MWOA and ALMWOA algorithms are tested on a set of 23 benchmark functions and the results are compared with standard WOA and other well-known meta-heuristic optimisation algorithms. Experimental results show that MWOA and ALMWOA can significantly outperform other optimisation algorithms in most of benchmark functions.

Keywords: whale optimisation algorithm; WOA; benchmark functions; meta-heuristic optimisation algorithms; Laplace crossover.

Reference to this paper should be made as follows: El Bakrawy, L.M. (2020) ‘An adaptive optimisation algorithm based on modified whale optimisation algorithm and Laplace crossover’, *Int. J. Metaheuristics*, Vol. 7, No. 3, pp.284–305.

Biographical notes: Lamiaa M. El Bakrawy received her BSc with honours in 2001 at the Pure Mathematics and Computer Science, Faculty of Science, Al-Azhar University, Egypt. She received her MSc and PhD in Computer Science from the Faculty of Science, Al-Azhar University, Egypt. She is currently working as a Doctor at the Faculty of Science, Al-Azhar University in Egypt. Her master topic was optimisation using swarm intelligence. Her PhD topic was machine learning in image authentication. Her research interests are image processing, machine learning, meta-heuristic optimisation strategies and information security.

1 Introduction

Many optimisation problems are very complex and NP-hard that need a lot of computational efforts to solve them. Conventional optimisation algorithms are often trapped in local optima when the problem search space increases. Compared to conventional optimisation algorithms, meta-heuristic algorithms have superior abilities in avoiding local optima. The stochastic nature of meta-heuristics algorithms helps them to search the whole search space and bypass stagnation in local solutions especially in real problems which are very complex with multiple local optima (Nesmachnow, 2014; Xu, 2013; Mirjalili and Lewis, 2014).

Currently, it has been observed that meta-heuristic optimisation algorithms which are inspired from nature have achieved a lot of attention in scientific engineering due to their flexibility, simplicity, robustness, and efficiency. They mimic various natural phenomena such as social behaviour of birds (Kennedy, 1995) behaviour of bats (Yang, 2010), pheromone of ants (Socha and Dorigo, 2008), behaviour of fire flies (Yang, 2013) and behaviour of wolves (Mirjalili and Lewis, 2014).

Mirjalili and Lewis (2016) presented a novel meta-heuristic optimisation algorithm, which mimics the hunting behaviour of humpback whale. Based on nature of humpback whale, they proposed whale optimisation algorithm (WOA). The results proved that WOA was very competitive with other meta-heuristic optimisation algorithms. Oliva et al. (2016) used improved chaotic WOA to detect the best configuration to the parameters of photovoltaic cells. Touma (2016) utilised WOA to find the optimal solution for the economic dispatch problem and tested the algorithm on system of IEEE 30- Bus. Prakash and Lakshminarayana (2016) used WOA to determine the optimal sizing and siting of capacitors in distribution network system. Ladumor et al. (2016) used WOA to solve the problem of unit commitment. Kaveh and Ghazaan (2016) applied an enhanced WOA for sizing optimisation problems of frame and truss structures. Mafarja and Mirjalili (2017) utilised hybrid WOA with simulated annealing to select the most informative features for classification task. Nasiri and Khiyabani (2018) used WOA algorithm for clustering. Hassan and Hassanien (2018) used WOA to extract the vasculature of retinal fundus images. Abdel-Basset et al. (2019) utilised modified WOA for solving single and multi-dimensional 0–1 knapsack problems. Bui et al. (2019) used WOA to select the optimal features and adjusting parameters of the adaptive neuro-fuzzy inference system for land-cover classification. Yousri et al. (2019) introduced four different variants of chaotic WOA and tested on CEC 2017 benchmark functions.

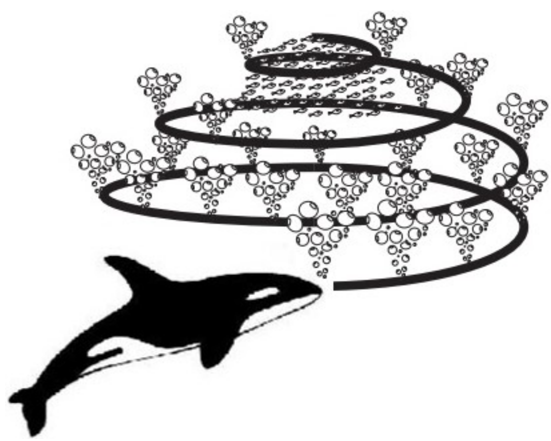
Standard WOA is easily trapped in local optima, provide slow convergence rate and lack of diversity, as the dimension of the search space expansion (Sun et al., 2019). In this paper, modified whale optimisation algorithm (MWOA) is proposed to improve the quality of standard WOA algorithm performance. In MWOA, the spiral movement of humpback whales is done based on archimedes' spiral to mimic the spiral position update of the humpback whales. Moreover, ALMWOA algorithm is presented in this paper to improve the local search capability of MWOA by increasing the diversity of search space. ALMWOA is based on the incorporation of MWOA and Laplace crossover. The proposed MWOA and ALMWOA algorithms are tested on a set of 23 benchmark functions (unimodal optimisation functions, multimodal optimisation functions and fixed-dimension multimodal optimisation functions). Experimental results demonstrate that the proposed algorithms are effective and have superior capabilities in most of benchmark functions.

The remainder of this paper is organised as follows: brief introduction of WOA and Laplace crossover are presented, respectively, in Sections 2 and 3. Section 4 describes in details the proposed MWOA and ALMWOA algorithms. Section 5 provides experimental results and analysis. At last, conclusions are discussed in Section 6.

2 Whale optimisation algorithm

WOA (Mirjalili and Lewis, 2016) is a new meta-heuristic optimisation algorithm which can be applied for solving optimisation problems. The algorithm simulates the hunting behaviour of humpback whales in searching and attacking preys (fish herds). WOA is inspired from the bubble-net hunting strategy of humpback whales, in which the whales dive approximately 12 metres down and start generating bubbles in a spiral shape around the fish herds. As shown in Figure 1, this strategy directs the fish herds towards the surface. Humpback whales attack the fish herds when they very close to the surface of the water. The mathematical modelling of hunting behaviour of humpback whales involves three various phases: encircling the prey phase, exploitation phase (bubble-net attacking) and exploration phase (searching for the prey).

Figure 1 Bubble-net attacking strategy of humpback whales



2.1 Encircling the prey phase

In this phase, humpback whales can notice the locations of the prey (fish herds), then they encircle them. In this algorithm, the current best position is the location of target prey that is closest to the perfect answer. After that, the best hunting agent (search agent) will be defined, and the rest of the candidate solutions (the other whales) will try to update their positions with reference to the location of the best hunting agent. The whole process is mathematically formulated as follows:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t_{cur}) - \vec{X}(t_{cur}) \right| \tag{1}$$

$$\vec{X}(t_{cur}+1) = \vec{X}^*(t_{cur}) - \vec{A} \cdot \vec{D} \tag{2}$$

where t_{cur} implies the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}^*$ denotes the position vector of the best or fittest solution acquired until this phase, $\vec{X}$ is the current position vector, $|\ |$ denotes the absolute value and $\cdot$ is a pairwise multiplication between two vectors. During every iteration, $\vec{X}^*$ should be updated if there is a better solution. The values of $\vec{A}$ and $\vec{C}$ coefficient vectors are computed as follows:

$$\vec{A} = 2\vec{au} \cdot \vec{r_1} - \vec{au} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r_1} \tag{4}$$

where $\vec{au}$ is search direction matrix linearly decreased from 2 to 0, over the algorithm iterations and $\vec{r_1}$ is a random vector value in the range [–1, 1].

2.2 Exploitation phase (bubble-net attacking)

The following two mechanisms are designed to mathematically model the exploitation phase (bubble-net attacking):

1 Shrinking encircling mechanism: This behaviour is accomplished by decreasing the value of $\vec{au}$ from 2 to 0 linearly via equation (3), over the algorithm iterations making $\vec{A}$ to have a random numbers in the range [–1, 1]. The new position of search agent can be defined any location from the initial position of the agent to the position of the current best agent.

2 Spiral updating mechanism: To formulate this behaviour, this mechanism calculates the distance between the current humpback whale position and the prey. Then, to mimic helix-shaped movement of humpback whales, a spiral equation is created between its current position and prey position as is given below:

$$\vec{X'}(t_{cur}+1) = \vec{D'} \cdot e^{b \cdot r_2} \cdot \cos(2\pi r_2) + \vec{X}^*(t_{cur}) \tag{5}$$

Here,

$$\vec{D'} = \left| \vec{X}^*(t_{cur}) - \vec{X}(t_{cur}) \right| \tag{6}$$

where $\vec{D'}$ is the distance from the current whale to the prey, b represents a constant parameter for defining the spiral movement shape by the whales, r_2 is a random value in the range [–1, 1] and $\cdot$ is a pairwise multiplication between two vectors. During optimisation process, it is noticed that the Humpback whales swim within shrinking encircling mechanism around the prey and following a spiral path toward the prey concurrently. For simplicity, to update the position of whales, the probability of 50% is assumed to choose between the shrinking encircling and the spiral updating mechanism. The mathematical model of these mechanisms can be expressed as follows:

Every whale may choose between spiral and shrinking position updates

$$\vec{X}(t_{cur}+1) = \begin{cases} \vec{X}^*(t_{cur}) - \vec{A} \cdot \vec{D} & \text{if } N_r < 0.5 \\ \vec{D'} \cdot e^{b \cdot r_2} \cdot \cos(2\pi r_2) + \vec{X}^*(t_{cur}) & \text{if } N_r \geq 0.5 \end{cases} \tag{7}$$

where N_r is a random number in the range [0, 1].

2.3 Exploration phase (searching for the prey)

This exploration phase (searching for the prey) is done based on the variation of vector $\vec{A}$ like the exploitation phase. In exploration phase, the value of vector $\vec{A}$ is greater than 1 or less than –1 to emphasise the search agents to swim far away from a reference whale. $| \vec{A} | > 1$ is used to force exploration in the WOA algorithm to make global search and bypass stagnation in local solutions. Exploration phase is distinct from exploitation phase in that the agent position in exploration phase is updated according to a randomly chosen agent instead of the best agent that has been computed so far. This phase is mathematically formulated as follows

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{rand}(t_{cur}) - \vec{X}(t_{cur}) \right| \tag{8}$$

$$\vec{X}(t_{cur} + 1) = \vec{X}_{rand}(t_{cur}) - \vec{A} \cdot \vec{D} \tag{9}$$

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

3 Laplace crossover

Deep and Thakur (2007) introduced a new crossover operator based on Laplace distribution. This operator is a parent centric real coded crossover which is known as Laplace crossover (LX). It generates two off-springs $y_1 = (y_{11}, y_{12}, ..., y_{1n})$ and $y_2 = (y_{21}, y_{22}, ..., y_{2n})$ from two parents $x_1 = (x_{11}, x_{12}, ..., x_{1n})$ and $x_2 = (x_{21}, x_{22}, ..., x_{2n})$. LX creates a distributed random number Q_i based on Laplace distribution according to the following equation:

$$Q_i = \begin{cases} l - klog_e(s_i) & s_i \leq 0.5 \\ l + klog_e(s_i) & s_i > 0.5 \end{cases} \tag{10}$$

where s_i is a uniformly distributed random number in the range [0, 1], $l \in R$ represents the location parameter and $k > 0$ is called the scale parameter. The offsprings are created according to the following equations:

$$y_{1i} = x_{1i} + Q_i \mid x_{1i} - x_{2i} \mid \tag{11}$$

$$y_{2i} = x_{2i} + Q_i \mid x_{1i} - x_{2i} \mid \tag{12}$$

Assume that $x_{\min i}$ and $x_{\max i}$ are minimum and maximum bounds of x_i. If the value of x_i is less than $x_{\min i}$ or greater than $x_{\max i}$, then x_i will be a random value in the range $[x_{\min i}, x_{\max i}]$. When k is a large value, LX produces the offspring distant from the parents. However when k is a small value, it produces the offspring close to the parents. LX circulates offsprings depending on the spread of parents if l and k are constant values.

4 The proposed algorithms

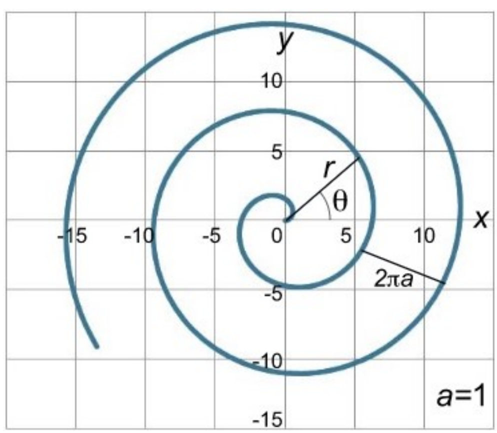
In this section, two modified WOA algorithms are proposed. MWOA is proposed to enhance convergence rate of standard WOA algorithm and avoid being stuck at local optimal solutions while ALMWOA algorithm is proposed to increase the diversity of search space and enhance the capability to avoid local optimal solutions. The details of the proposed algorithms will be described as follows.

4.1 Modified whale optimisation algorithm

In standard WOA algorithm, a spiral equation between the positions of humpback whale and target prey is created to mimic the helix-shaped movement according to equation (5). The humpback whales move with a logarithmic spiral function to attack the prey and cover the border area in the search space. WOA algorithm suffers from falling into local optima, especially for solving the high dimension optimisation problems.

The core of our proposed algorithm (MWOA) aims to employ archimedes' spiral and variation of vector $\vec{A}$ to mimic the spiral position update of the humpback whales over the algorithm iterations. Archimedes' spiral is a kind of an archimedean (arithmetic) spiral. It is expressed by polar equation in the form $r = a\theta$ in which r and θ are the length of the radius from the origin of the spiral and the polar angle (the angular position of the radius), respectively, and a is a constant. In archimedes' spiral, the distance between successive turnings is constant due to the linear relation between radius and the angle as shown in Figure 2.

Figure 2 Archimedes' spiral in polar coordinate system (see online version for colours)



The main difference between archimedean spiral and logarithmic spiral is that archimedean spiral has a beginning from the origin of the spiral, while the logarithmic tends towards the origin. Thus in MWOA algorithm, the humpback whales move with archimedes' spiral function to attack the target prey by the following equation:

$$\vec{X}'(t_{cur} + 1) = \vec{D}' \cdot b \cdot r_2 \cdot \cos(2\pi r_2) + \vec{A} \cdot \vec{X}^*(t_{cur}) \qquad (13)$$

The main steps of MWOA are depicted in Algorithm 1.

Algorithm 1 Pseudo code of MWOA algorithm

Set number of whales = N
Set maximum number of iterations = Max_Iter
Initialise the whales population X_i (i = 1, 2, ..., N)
Initialise A, C and r_2
Evaluate the fitness function of each whale
Set X^* = the position of best whale based on fitness function
Set t_{cur} = 0
  while $t_{cur} < Max\_Iter$ **do**
    for $i = 1$ to N **do**
      Update A, C and r_2
      Generate randomly $N_r \in$ [0, 1]
      if (N_r < 0.5) **then**
        if $\mid A \mid < 1$ **then**
          $\vec{D} = \left|\vec{C} \cdot \vec{X}^*(t_{cur}) - \vec{X}(t_{cur})\right|$
          $\vec{X}(t_{cur}+1) = \vec{X}^*(t_{cur}) - \vec{A} \cdot \vec{D}$
        else if $\mid A \mid \geq 1$ **then**
          $\vec{D} = \left|\vec{C} \cdot \vec{X}_{rand}(t_{cur}) - \vec{X}(t_{cur})\right|$
          $\vec{X}(t_{cur}+1) = \vec{X}_{rand}(t_{cur}) - \vec{A} \cdot \vec{D}$
        end if
      else if ($N_r \geq$ 0.5) **then**
        $\vec{D'} = \left|\vec{X}^*(t_{cur}) - \vec{X}(t_{cur})\right|$
        $\vec{X'}(t_{cur}+1) = \vec{D'} \cdot b \cdot r_2 \cdot \cos(2\pi r_2) + \vec{A} \cdot \vec{X}^*(t_{cur})$
      end if
    end for
    Evaluate the fitness function of all whales
    Update the value of X^* based on fitness function
    $t_{cur} = t_{cur} + 1$
  end while
Return the best solution X^*

4.2 ALMWOA algorithm

To enhance MWOA performance, an adaptive optimisation algorithm based on MWOA and Laplace crossover is proposed.

In the ALMWOA algorithm, MWOA has been incorporated with Laplace crossover. At each iteration, ALMWOA algorithm simulates the MWOA first, then Laplace crossover is applied to two search agents (whales). The first agent is the current best whale while the second agent is chosen randomly from the current population. Laplace crossover generates two off-springs according to equations (11) and (12) respectively and the fitness function (F) of them are compared with fitness function of the current worst whale one by one. If the fitness function of any one of two off-springs is better than the fitness function of the current worst whale, then the current worst whale is substituted by better off-spring. The fitness function of two off-springs are also compared with fitness function of the current best whale one by one. If the fitness function of any one of two off-springs is better than the fitness function of the current best whale, then the current best whale is substituted by better off-spring. This procedure is repeated until the termination criterion for iterations in ALMWOA algorithm is satisfied according to whether the maximum number of iterations or minimum fitness function error is reached. The main steps of ALMWOA algorithm are shown in Algorithm 2.

Algorithm 2 Pseudo code of ALMWOA algorithm

Set number of whales = N
Set maximum number of iterations = Max_Iter
Initialise the whales population X_i (i = 1, 2, ..., N)
Initialise A, C and r_2
Evaluate the fitness function of each whale
Set $X^* =$ the position of best whale based on fitness function
Set t_{cur} = 0
  while $t_{cur} < Max\_Iter$ **do**
    for $i = 1$ to N **do**
      Update A, C and r_2
      Generate randomly $N_r \in$ [0, 1]
      if ($N_r < 0.5$) **then**
        if $| A |< 1$ **then**
          $\vec{D} = \left|\vec{C} \cdot \vec{X}^*(t_{cur}) - \vec{X}(t_{cur})\right|$
          $\vec{X}(t_{cur} + 1) = \vec{X}^*(t_{cur}) - \vec{A} \cdot \vec{D}$
        else if $| A |\geq 1$ **then**
          $\vec{D} = \left|\vec{C} \cdot \vec{X}_{rand}(t_{cur}) - \vec{X}(t_{cur})\right|$
          $\vec{X}(t_{cur} + 1) = \vec{X}_{rand}(t_{cur}) - \vec{A} \cdot \vec{D}$
        end if
      else if ($N_r \geq 0.5$) **then**
        $\vec{D'} = \left|\vec{X}^*(t_{cur}) - \vec{X}(t_{cur})\right|$
        $\vec{X}'(t_{cur} + 1) = \vec{D'} \cdot b \cdot r_2 \cdot \cos(2\pi r_2) + \vec{A} \cdot \vec{X}^*(t_{cur})$
      end if
    end for
    Evaluate the fitness function of all whales
    Update the value of X^* based on fitness function
    Set $x_1 = X^*$ and $x_2 =$ random search agent (whale)
    Apply laplace crossover
    Generate two off _ springs y_1 and y_2
    Set X^w the position of worst whale based on fitness function
    if F(y_1) better than F(X^w) **then**
      $X^w = y_1$
    else if F(y_2) better than F(X^w) **then**
      $X^w = y_2$
    end if
    if F(y_1) better than F(X^*) **then**
      $X^* = y_1$
    end if
    if F(y_2) better than F(X^*) **then**
      $X^* = y_2$
    end if
    $t_{cur} = t_{cur} + 1$
  end while
Return the best solution X^*

---

5 Experimental results and analysis

To verify the performance of MWOA and ALMWOA algorithms, 23 benchmark functions are used. The benchmark functions are divided into three types: unimodal optimisation functions (Table 1), multimodal optimisation functions (Table 2), and fixed-dimension multimodal optimisation functions (Table 3). Where Dim represents the dimension of variables, Range denotes the boundary of the function's search space, and $F_{\min}$ is the optimum of the functions. Figure 3 illustrates the typical 2D plots of the benchmark functions for F_1, F_5, F_{16} and F_{23} test cases used in this paper.

Table 1 Description of unimodal optimisation functions

Name	*Function*	*Dim*	*Range*	$F_{\min}$
Sphere	$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[–100, 100]	0
Schwefel 2.22	$F_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[–10, 10]	0
Schwefel 1.2	$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[–100, 100]	0
Schwefel 2.21	$F_4(x) = \max_i\{\lvert x_i\rvert, 1 \leq i \leq^n\}$	30	[–100, 100]	0
Rosenbrock	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[–30, 30]	0
Step	$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[–100, 100]	0
Noise	$F_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	30	[–1.28, 1.28]	0

Table 2 Description of multimodal optimisation functions

Name	*Function*	*Dim*	*Range*	$F_{\min}$
Schwefel 2.26	$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	30	[–500, 500]	–418.9829 × 5
Rastrigin	$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[–5.12, 5.12]	0
Ackley	$F_{10}(x) = - - 20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$ $- \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[–32, 32]	0
Griewank	$F_{11}(x) = \frac{1}{4{,}000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[–600, 600]	0
Penalise 1	$F_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1)$ $+ \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m)$ $= \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[–50, 50]	0
Penalise 2	$F_{13}(x) = 0.1\{\sin^2(3\pi x_1)$ $+ \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)]$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[–50, 50]	0

The experiments are performed on a computer with 2.40 GHZ frequency, Intel(R) Core(TM) i7-5500U central processing unit (CPU) and 16 GB random-access memory (RAM) using written codes in MATLAB R2015a. In these experiments, the MWOA and ALMWOA algorithms are run thirty independent times for each benchmark function with population size of the whales $N = 30$ and maximum number of iterations $Max\_Iter = 500$. The location parameter of Laplace crossover is $l = 0$ and the scale parameter of Laplace crossover is $k = 0.1$ (after extensive experimentation, these parameters are fine tuned).

Table 3 Description of fixed-dimension multimodal optimisation functions

Name	*Function*	*Dim*	*Range*	$F_{\min}$
De Jong	$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	[–65, 65]	1
Kowalik	$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[–5, 5]	0.00030
Camel Back-6 Hump	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[–5, 5]	–1.0316
Branin	$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[–5, 5]	0.398
Goldstein-price	$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[–2, 2]	3
Hartman 3	$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1, 3]	–3.86
Hartman 6	$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	–3.32
Shekel 1	$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	–10.1532
Shekel 2	$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	–10.4028
Shekel 3	$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	–10.5363

Figure 3 Typical 2D representations of some benchmark functions (see online version for colours)

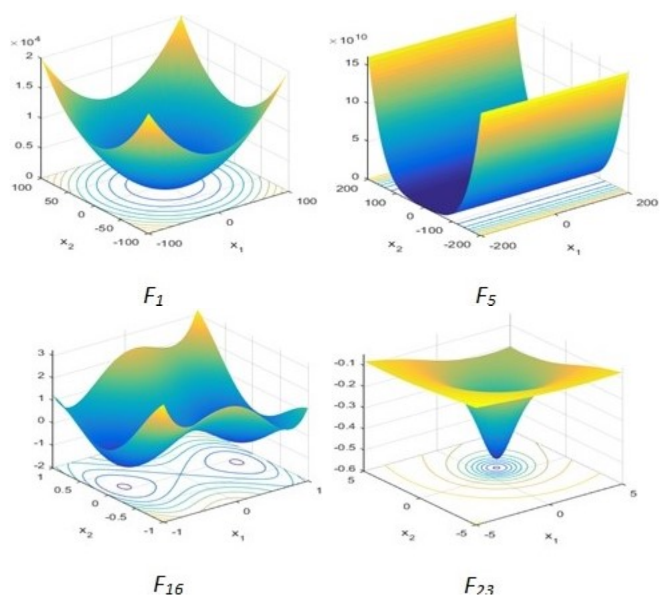


Table 4 Results of the p-value for t-test on benchmark functions at 0.05 level of significance with 95% confidence interval

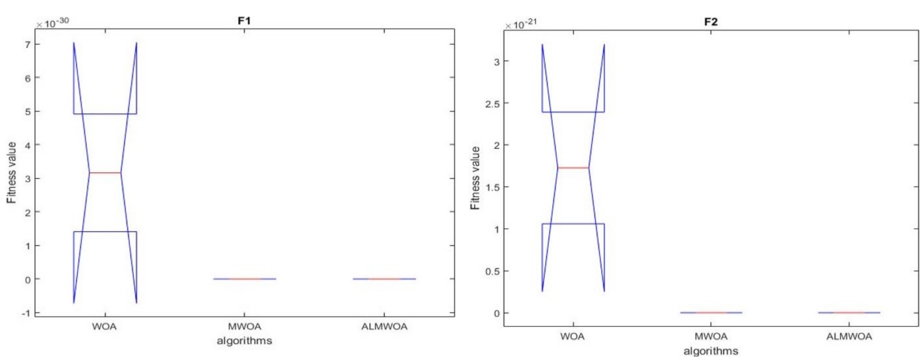
Function	*WOA vs. MWOA*	*WOA vs. ALMWOA*
F1	0.0443	0.0443
F2	0.0295	0.0295
F3	1.3315E-17	1.3315E-17
F4	2.0793E-09	2.0793E-09
F5	9.4436E-05	1.1717E-06
F6	1.0814E-04	1.1420E-04
F7	1.3376E-05	1.9491E-05
F8	9.3224E-08	1.6315E-07
F9	0.1608	0.1608
F10	4.1883E-10	4.1883E-10
F11	0.3256	0.3256
F12	1.5380E-07	0.0041
F13	2.0534E-11	0.0032
F14	0.0358	0.0115
F15	0.0237	0.0011
F16	4.0695e-04	0.0263
F17	0.0603	0.0336
F18	0.0281	0.0234
F19	0.0057	0.0158
F20	0.0041	0.0283
F21	0.0039	3.4917e-04
F22	7.0881E-07	1.4194E-08
F23	0.0480	9.9388E-04

For verifying the results of the proposed algorithms, MWOA and ALMWOA are compared against standard WOA (Mirjalili and Lewis, 2016), particle swarm optimisation (PSO) (Kennedy, 1995), gravitational search algorithm (GSA) (Rashedi et al., 2009), and differential evolution (DE) (Storn and Price, 1997). The results of the comparative algorithms are taken from Mirjalili and Lewis (2016) in terms of AV and SD. Moreover, t-test performance of the proposed algorithms, MWOA and ALMWOA are compared with WOA using a pairwise one tailed t-test at 0.05 level of significance over the fitness values of all optimisation functions considered. t-test should be performed to determine whether two samples from a normal distribution could have the same mean or not (Sun et al., 2014). The null hypothesis is assumed that the means of the two algorithms are equal at 0.05 level of significance and alternative hypothesis is assumed that the means of the two algorithms are different. p-value is the significance associated with t-test with varying levels of evidence: a p-value greater than 0.1 implies 'not significant'; a p-value less than 0.1 but greater than 0.05 yields 'marginally significant'; a p-value less than or equal to 0.05 but greater than 0.01 yields 'significant'; and a p-value less than 0.01 constitutes 'highly significant' (Filho et al., 2013).

Table 4 shows the results of the p-value for t-test on benchmark functions at 0.05 level of significance with 95% confidence interval. The results of Table 4 show that if WOA vs. MWOA is considered then 14 out of the 23 problems illustrate that MWOA is highly significant than WOA, 6 out of the 23 problems illustrate that MWOA is significantly better than WOA, 1 out of the 23 problems illustrate that MWOA is marginally significant than WOA and 2 out of the 23 problems illustrate that MWOA is not significant. If WOA vs. ALMWOA is considered then 13 out of the 23 problems show that ALMWOA is highly significant than WOA, 8 out of the 23 problems illustrate that MWOA is significantly better than WOA and 2 out of the 23 problems illustrate that ALMWOA is not significant.

The unimodal optimisation functions ($F_1 - F_7$) are suitable for examining the exploitation and convergence of an optimisation algorithm since they have only one global optimum. Figures 4–7 provide the anova test for the fitness values of WOA, MWOA and ALMWOA for unimodal optimisation functions. According to Table 5 and Figures 4–7, it is observed that the performance of MWOA and ALMWOA algorithms is better than standard WOA for the majority of the test cases. In particular, MWOA algorithm shows better performance than standard WOA in terms of average AV for functions $F_1 - F_4$ and $F_6 - F_7$. For the seven functions on unimodal optimisation functions, the standard deviation SD of MWOA algorithm is also less than standard WOA, which proves that applying archimedes' spiral to mimic the spiral updating position can enhance convergence rate. Moreover, MWOA is the best efficient optimiser for function F_7 compared to standard WOA, ALMWOA, PSO, GSA and DE. Also MWOA has the best SD for functions F_1 and F_3 similar to ALMWOA. While given the second rank for functions F_1, F_2 and F_3 in terms of AV among other comparative algorithms. The results also indicate that ALMWOA has performance improvement over WOA and MWOA in terms of AV for functions $F_1 - F_7$ and $F_1 - F_6$ respectively. ALMWOA has better SD than WOA and MWOA for functions $F_1 - F_7$ and $F_1 - F_4$ respectively. Table 5 indicates that ALMWOA is the best efficient optimiser for functions F_1, F_2 and F_3 compared to standard WOA, MWOA, PSO, GSA and DE and the second rank for functions F_4, F_5 and F_7 in terms of AV. The better performance of ALMWOA is related to applying archimedes' spiral with Laplace crossover. Hence, the ALMWOA algorithm can provide very good exploitation and convergence behaviour.

Figure 4 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_1 and F_2 (see online version for colours)



Unlike unimodal optimisation functions, multimodal optimisation functions ($F_8 - F_{13}$) possess a large number of local optima whose number increases exponentially as the expansion of the problem size. Therefore, this kind of optimisation problems are suitable for examining the exploration ability of an optimisation algorithm. fixed-dimension multimodal optimisation functions ($F_{14} - F_{23}$) are similar to multimodal optimisation functions, since they have more than one local optima but the main difference between them is that they provide different search space compared to multimodal optimisation functions.

Figure 5 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_3 and F_4 (see online version for colours)

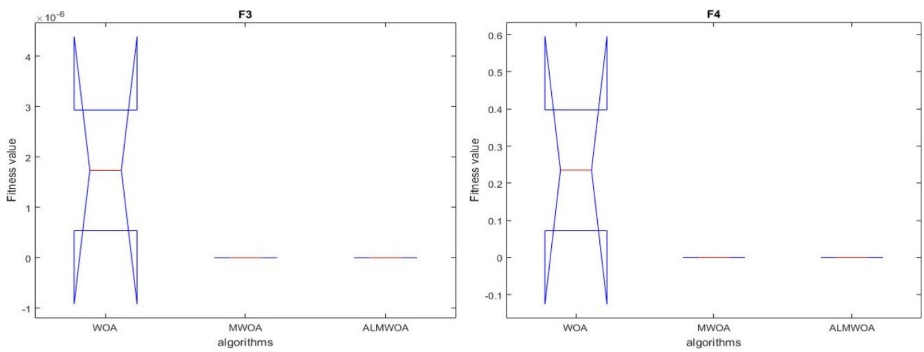


Figure 6 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_5 and F_6 (see online version for colours)

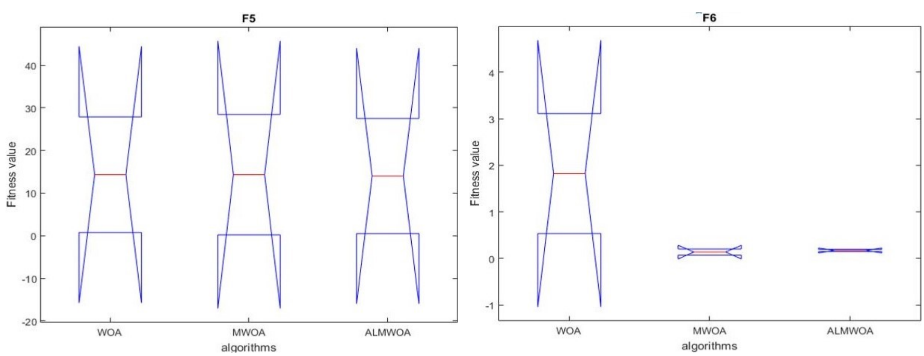


Figure 7 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_7 (see online version for colours)

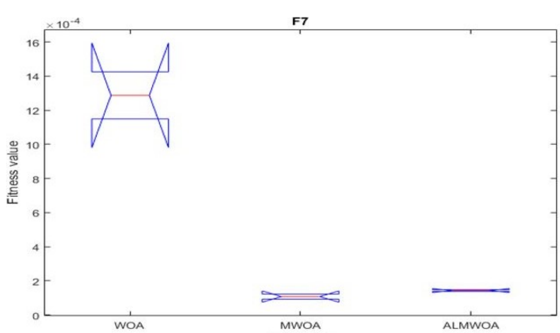


Table 5 Optimisation results obtained for unimodal optimisation functions

Function	*WOA*		*MWOA*		*ALMWOA*		*PSO*		*GSA*		*DE*	
	AV	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*
F1	1.41E-30	4.91E-30	5.5406E-268	*0*	*0*	*0*	0.000136	0.000202	2.53E-16	9.67E-17	8.2E-14	5.9E-14
F2	1.06E-21	2.39E-21	2.2464E-139	7.6660E-139	*1.2014E-200*	*0*	0.042144	0.045421	0.055655	0.194074	1.5E-09	9.9E-10
F3	5.39E-07	2.93E-06	3.1291E-249	*0*	*0*	*0*	70.12562	22.11924	896.5347	318.9559	6.8E-11	7.4E-11
F4	0.072581	0.39747	8.2080E-139	3.8104E-138	6.8535E-190	*0*	1.086481	0.317039	7.35487	1.741452	*0*	*0*
F5	27.86558	0.763626	28.4454	0.1924	27.4978	0.5048	96.71832	60.11559	67.54309	62.22534	*0*	*0*
F6	3.116266	0.532429	0.2014	0.0691	0.1937	0.1451	0.000102	8.28E-05	2.5E-16	1.74E-16	*0*	*0*
F7	0.001425	0.001149	*1.2271E-04*	*9.2769E-05*	1.4830E-04	1.3800E-04	0.122854	0.044957	0.089441	0.04339	0.00463	0.0012

Figure 8 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_8 and F_{10} (see online version for colours)

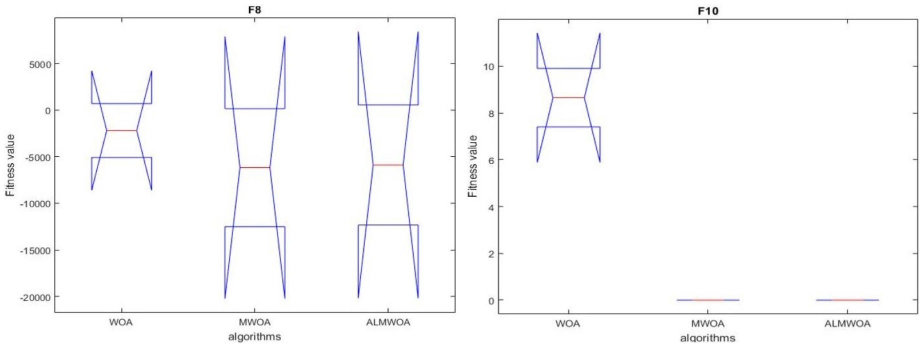


Figure 9 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_{11} and F_{12} (see online version for colours)

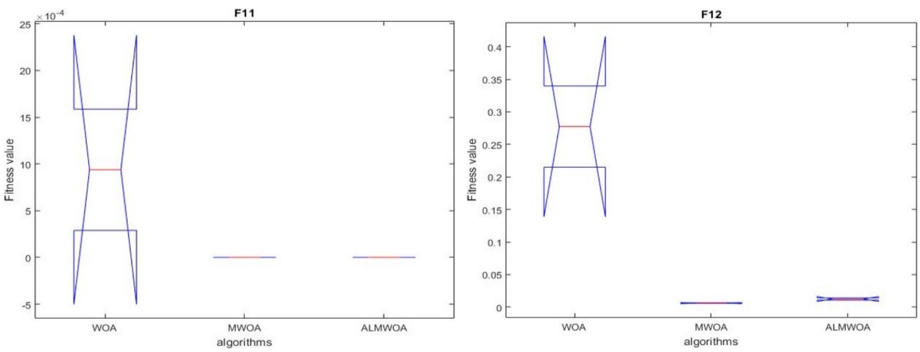
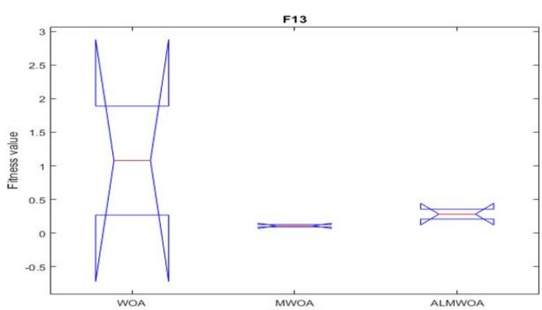


Figure 10 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_{13} (see online version for colours)



Figures 8–10 show the anova test for the fitness values of WOA, MWOA and ALMWOA for multimodal optimisation functions. From Table 6 and Figures 8–10, it is obvious that the performance of MWOA and ALMWOA algorithms is better than standard WOA, since MWOA algorithm has better AV and SD than standard WOA for functions F_8, $F_{10} - F_{13}$. Also MWOA algorithm has the same results to

WOA for function F_9. Results show that MWOA is the best efficient optimiser for functions $F_8 - F_{11}$ and given the second rank for functions F_{12} and F_{13} among other comparative algorithms. The results in Table 6 also indicate that ALMWOA has the best performance for function F_{10} and has the same results to WOA for function F_9 and similar to MWOA for functions F_9 and F_{11} which verify that both algorithms MWOA and ALMWOA have superior abilities in avoiding local optima for multimodal optimisation functions.

Figure 11 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_{14} and F_{15} (see online version for colours)

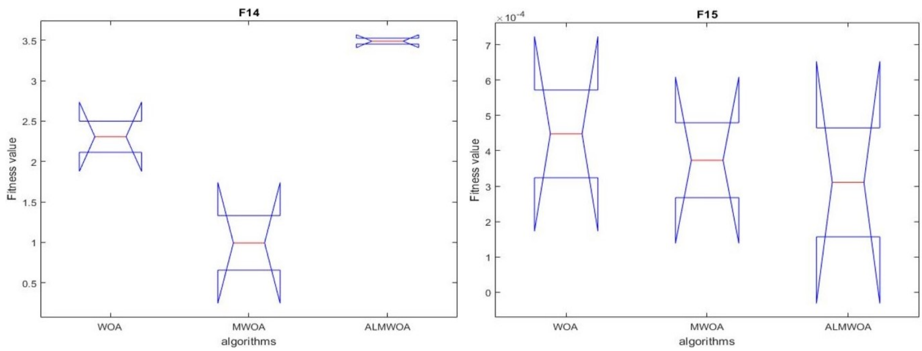
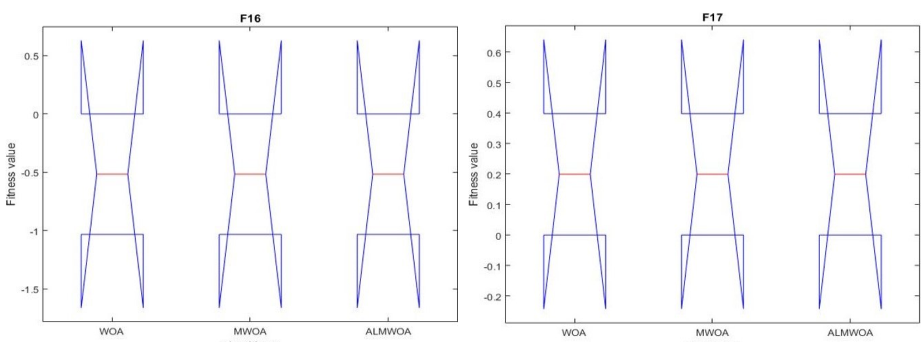


Figure 12 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_{16} and F_{17} (see online version for colours)



Figures 11–15 indicate the anova test for the fitness values of WOA, MWOA and ALMWOA for fixed-dimension multimodal optimisation functions. Table 7 and Figures 11–15 demonstrate the efficiency and superior of MWOA and ALMWOA algorithms than standard WOA in most cases. MWOA algorithm has better AV and SD than standard WOA for functions F_{14}, F_{15} and $F_{20} - F_{23}$ and has the same AV to WOA for F_{16}, F_{18} and F_{19}. Moreover, MWOA is the best efficient optimiser (AV) for functions $F_{16} - F_{18}$ and has the second rank (AV) for function F_{14} and the second rank (SD) for function F_{21} among all other algorithms. The results in Table 7 also show that ALMWOA has the best performance for functions $F_{18} - F_{20}$ and F_{23}, the best AV for

function F_{22} and the best SD for function F_{15}. While ALMWOA has the second best AV for functions F_{15} and F_{21} and the second best SD for function F_{22}. This is due to incorporated archimedes' spiral with Laplace crossover for exploration in the standard WOA algorithm that directs ALMWOA algorithm towards the global optimum. These results demonstrate that both algorithms MWOA and ALMWOA are potentially able to solve fixed-dimension multimodal optimisation functions.

Figure 13 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_{18} and F_{19} (see online version for colours)

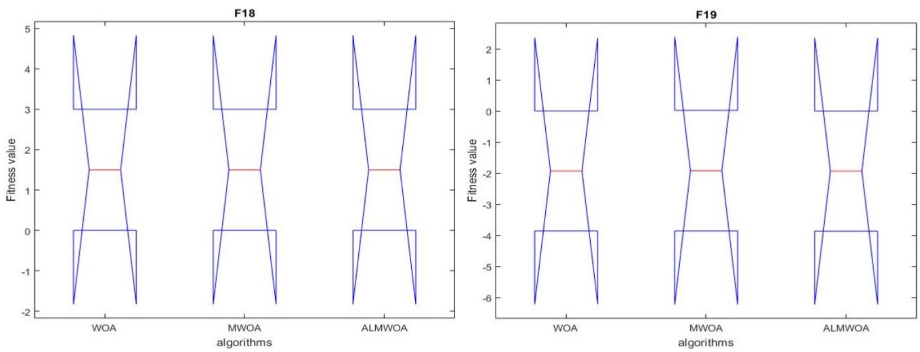


Figure 14 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_{20} and F_{21} (see online version for colours)

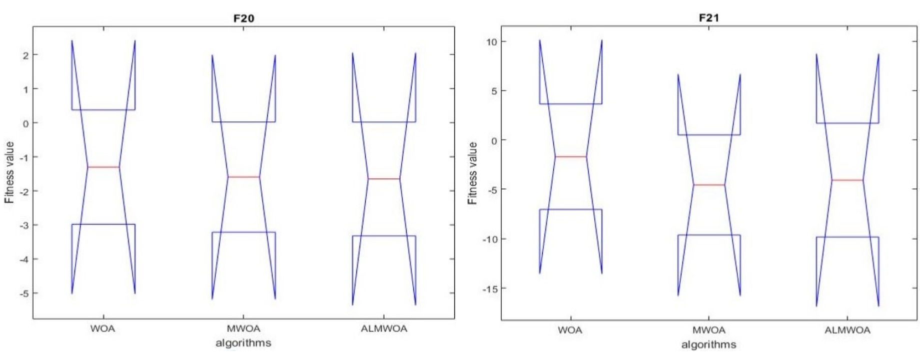


Figure 15 ANOVA test for the fitness values of WOA, MWOA and ALMWOA algorithms for F_{22} and F_{23} (see online version for colours)

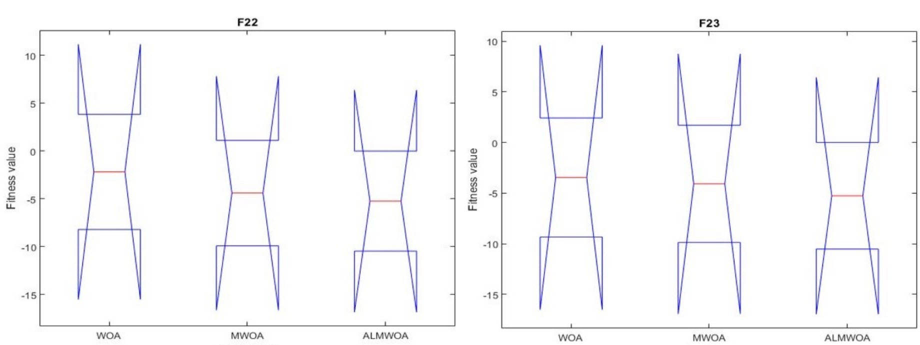


Table 6 Optimisation results obtained for multimodal optimisation functions

Function	*WOA*		*MWOA*		*ALMWOA*		*PSO*		*GSA*		*DE*	
	AV	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*
F8	–5,080.76	695.7968	*–12,512*	*161.6243*	–12,328	556.6175	–4,841.29	1,152.814	–2,821.07	493.0375	–11,080.1	574.7
F9	*0*	*0*	*0*	*0*	*0*	*0*	46.70423	11.62938	25.96841	7.470068	69.2	38.8
F10	7.4043	9.897572	*8.8818E-16*	*0*	*8.8818E-16*	*0*	0.276015	0.50901	0.062087	0.23628	9.7E-08	4.2E-08
F11	0.000289	0.001586	*0*	*0*	*0*	*0*	0.009215	0.007724	27.70154	5.040343	*0*	*0*
F12	0.339676	0.214864	0.0065	0.0057	0.0141	0.0107	0.006917	0.026301	1.799617	0.95114	*7.9E-15*	*8E-15*
F13	1.889015	0.266088	0.1247	0.0895	0.3537	0.2081	0.006675	0.008907	8.899084	7.126241	*5.1E-14*	*4.8E-14*

Table 7 Optimisation results obtained for fixed-dimension multimodal optimisation functions

Function	WOA		MWOA		ALMWOA		PSO		GSA		DE	
	AV	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*	*AV*	*SD*
F14	2.111973	2.498594	1.3290	0.6559	3.4513	3.5229	3.627168	2.560828	5.859838	3.831299	*0.998004*	*3.3E-16*
F15	0.000572	0.000324	0.00047946	0.00026748	0.00046488	*0.00015663*	0.000577	0.000222	0.003673	0.001647	*4.5E-14*	0.00033
F16	*–1.03163*	4.2E-07	*–1.03163*	1.1644E-06	*–1.03163*	5.5419E-08	*–1.03163*	6.25E-16	*–1.03163*	4.88E-16	*–1.03163*	*3.1E-13*
F17	0.397914	2.7E-05	*0.397887*	2.2693E-04	*0.397887*	2.5568E-06	*0.397887*	*0*	*0.397887*	*0*	*0.397887*	9.9E-09
F18	*3*	4.22E-15	*3*	5.5869E-04	*3*	*1.11E-15*	*3*	1.33E-15	*3*	4.17E-15	*3*	2E-15
F19	–3.85616	0.002706	–3.85616	0.0231	*–3.86278*	*0*	*–3.86278*	2.58E-15	*–3.86278*	2.29E-15	N/A	N/A
F20	–2.98105	0.376653	–3.2155	0.0220	*–3.323*	*0.018134*	–3.26634	0.060516	–3.31778	0.023081	N/A	N/A
F21	–7.04918	3.629551	–9.6294	0.4982	–9.8424	1.6926	–6.8651	3.019644	–5.95512	3.737079	*–10.1532*	*0.0000025*
F22	–8.18178	3.829202	–9.9024	1.1135	*–10.4544*	0.0029	–8.45653	3.087094	–9.68447	2.014088	–10.4029	*3.9E-07*
F23	–9.34238	2.414737	–9.8718	1.6969	*–10.5364*	*1.8157E-15*	–9.95291	1.782786	*–10.5364*	2.6E-15	*–10.5364*	1.9E-07

Figure 16 Convergence plots of WOA, MWOA and ALMWOA algorithms for some benchmark functions (see online version for colours)

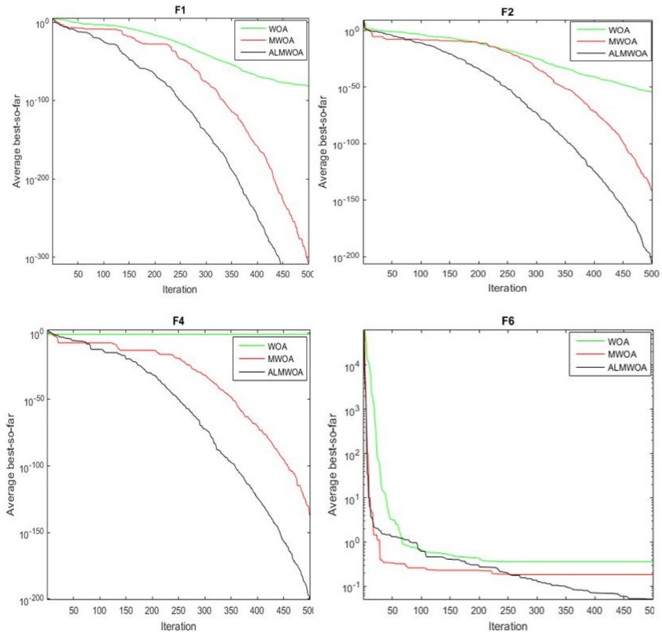
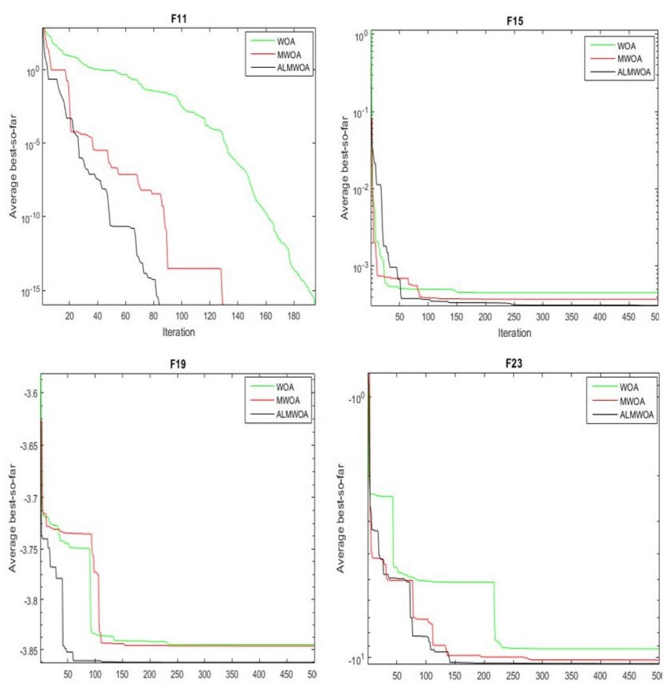


Figure 17 Convergence plots of WOA, MWOA and ALMWOA algorithms for some benchmark functions (see online version for colours)



In order to observe the behaviour of fitness values vs. the iterations, the convergence plots of WOA, MWOA and ALMWOA algorithms over the different runs are shown in Figures 16 and 17.

As can be observed, the MWOA and ALMWOA algorithms are superior to WOA algorithm but the ALMWOA algorithm is converging fast towards optima from the initial steps of iterations in comparison to WOA, MWOA algorithms. Hence, the ALMWOA algorithm gains from good exploration and exploitation, which consequently helps the ALMWOA algorithm to avoid being stuck at local optimal solutions.

6 Conclusions

This paper proposed two modified optimisation algorithms based on WOA algorithm called MWOA and ALMWOA. In the proposed MWOA algorithm, archimedes' spiral is adopted to mimic the spiral position update of the humpback whales to enhance convergence rate of standard WOA algorithm and avoid it of being stuck at local optimal solutions over the algorithm iterations. While, ALMWOA algorithm incorporate MWOA with Laplace crossover to insure the diversity of search space and improve the capability to avoid local optimal solutions. To evaluate the performance of these two proposed algorithms, 23 benchmark functions are employed and results are compared with WOA, PSO, GSA and DE. Experimental results illustrate that our proposed algorithms can provide highly competitive results in a majority of benchmark functions because of their fast convergence and little chance to get stuck at local optima. Our future work direction will be to apply our proposed algorithms to solve many engineering optimisation problems. In addition, future work can focus on developing new versions of WOA.

References

Abdel-Basset, M., El-Shahat, D. and Sangaiah, A.K. (2019) 'A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem', *International Journal of Machine Learning and Cybernetics*, Vol. 10, No. 3, pp.495–514.

Bui, Q.T., Pham, M.V., Nguyen, Q.H., Nguyen, L.X. and Pham, H.M. (2019) 'Whale optimization algorithm and adaptive neuro-fuzzy inference system: a hybrid method for feature selection and land pattern classification', *International Journal of Remote Sensing*, Vol. 40, No. 13, pp.1–16.

Deep, K. and Thakur, M. (2007) 'A new crossover operator for real coded genetic algorithms', *Applied Mathematics and Computation*, Vol. 188, No. 1, pp.895–911.

Filho, D., Paranhos, R., da Rocha, E., Batista, M., da Silva, J., Santos, M. and Marino, J. (2013) 'When is statistical significance not significant?', *Brazilian Political Science Review*, Vol. 7, No. 1, pp.31–55.

Hassan, G. and Hassanien, A.E. (2018) 'Retinal fundus vasculature multilevel segmentation using whale optimization algorithm', *Signal Image Video Process*, Vol. 12, No. 2, pp.263–270.

Kaveh, A. and Ghazaan, M.I. (2016) 'Enhanced whale optimization algorithm for sizing optimization of skeletal structures', *Mechanics Based Design of Structures and Machines*, Vol. 45, No. 3, pp.1–18.

Kennedy, J. (1995) 'Particle swarm optimization', *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, pp.1942–1948.

Ladumor, D., Trivedi, I., Jangir, P. and Kumar, A. (2016) 'A whale optimization algorithm approach for unit commitment problem solution', *Proceeding of the 2016 National Conference on Advancements in Electrical and Power Electronics Engineering*, pp.1–4.

Mafarja, M. and Mirjalili, S. (2017) 'Hybrid whale optimization algorithm with simulated annealing for feature selection', *Neurocomputing*, Vol. 260, pp.302–312.

Mirjalili, S. and Lewis, A. (2014) 'Grey wolf optimizerc', *Advances in Engineering Software*, Vol. 69, pp.46–61 [online] https://www.sciencedirect.com/science/article/abs/pii/S0965997813001853?via%3Dihub.

Mirjalili, S. and Lewis, A. (2016) 'The whale optimization algorithm', *Advances in Engineering Software*, Vol. 95, pp.51–67 [online] https://www.sciencedirect.com/science/article/abs/pii/S0965997816300163.

Nasiri, J. and Khiyabani, F. (2018) 'A whale optimization algorithm (WOA) approach for clustering', *Cogent Mathematics and Statistics*, Vol. 5, No. 1, pp.1–13.

Nesmachnow, S. (2014) 'An overview of metaheuristics: accurate and efficient methods for optimisation', *International Journal of Metaheuristics*, Vol. 3, No. 4, pp.320–346.

Oliva, D., Abd El Aziz, M. and Hassanien, A. (2017) 'Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm', *Applied Energy*, Vol. 200, pp.141–154 [online] https://inis.iaea.org/search/citationdownload.aspx.

Prakash, D.B. and Lakshminarayana, C. (2016) 'Optimal siting of capacitors in radial distribution network using whale optimization algorithm', *Alexandria Engineering Journal*, Vol. 56, No. 4, pp.499–509.

Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. (2009) 'GSA: a gravitational search algorithm', *Information Science*, Vol. 179, No. 13, pp.2232–2248.

Socha, K. and Dorigo, M. (2008) 'Ant colony optimization for continuous domains', *European Journal of Operational Research*, Vol. 185, No. 3, pp.1155–1173.

Storn, R. and Price, K. (1997) 'Differential evolutiona a simple and efficient heuristic for global optimization over continuous spaces', *Journal of Global Optimization*, Vol. 11, No. 4, pp.341–359.

Sun, Y., Djouani, K., Jacobus, V.B., Wang, Z. and Siarry, P. (2014) 'Hypothesis testing-based adaptive PSO', *Journal of Engineering, Design and Technology*, Vol. 12, No. 1, pp.89–101.

Sun, Y., Yang, T. and Liu, Z. (2019) 'A whale optimization algorithm based on quadratic interpolation for high-dimensional global optimization problems', *Applied Soft Computing Journal*, Vol. 85, pp.1–20 [online] https://www.sciencedirect.com/science/article/abs/pii/S1568494619305253?via%3Dihub.

Touma, H.J. (2016) 'Study of the economic dispatch problem on IEEE 30-bus system using whale optimization algorithm', *International Journal on Engineering Technology and Sciences*, Vol. 5, No. 1, pp.11–18.

Xu, G. (2013) 'An adaptive parameter tuning of particle swarm optimization algorithm', *Applied Mathematics and Computation*, Vol. 219, No. 9, pp.4560–4569.

Yang, X.S. (2010) 'A new metaheuristic bat-inspired algorithm', *Nature Inspired Cooperative Strategies for Optimization*, Vol. 284, pp.65–74 [online] https://link.springer.com/chapter/10.1007/978-3-642-12538-6_6.

Yang, X.S. (2013) 'Multi-objective firefly algorithm for continuous optimization', *Engineering with Computers*, Vol. 29, No. 2, pp.175–184.

Yousri, D., Allam, D. and Eteiba, M.B. (2019) 'Chaotic whale optimizer variants for parameters estimation of the chaotic behavior in permanent magnet synchronous motor', *Applied Soft Computing*, Vol. 74, pp.479–503 [online] https://www.researchgate.net/publication/328677785_Chaotic_whale_optimizer_variants_for_parameters_estimation_of_the_chaotic_behavior_in_Permanent_Magnet_Synchronous_Motor.