FUZZY ADAPTIVE WHALE OPTIMIZATION ALGORITHM FOR NUMERIC OPTIMIZATION

Ersin Kaya[1]*, Alper Kılıç[2], İsmail Babaoğlu[3] and Ahmet Babalık[4]

[1,2,3,4]Department of Computer Engineering, Konya Technical University,
Konya, Turkey

Email: ekaya@ktun.edu.tr[1*] (corresponding author), akilic@ktun.edu.tr[2], ibabaoglu@ktun.edu.tr[3], ababalik@ktun.edu.tr[4]

DOI: https://doi.org/10.22452/mjcs.vol34no2.4

ABSTRACT

Meta-heuristic approaches are used as a powerful tool for solving numeric optimization problems. Since these problems are deeply concerned with their diversified characteristics, investigation of the utilization of algorithms is significant for the researchers. Whale optimization algorithm (WOA) is one of the novel meta-heuristic algorithms employed for solving numeric optimization problems. WOA deals with exploitation and exploration of the search space in three stages, and in every stage, all dimensions of the candidate solutions are updated. The drawback of this update scheme is to lead the convergence of the algorithm to stack. Some known meta-heuristic approaches treat this issue by updating one or a predetermined number of dimensions in their update scheme. To improve the exploitation behavior of WOA, a fuzzy logic controller (FLC) based adaptive WOA (FAWOA) is suggested in this study. An FLC realizes the update scheme of WOA, and the proposed FLC determines the rate of the change in terms of dimension. The suggested FAWOA is evaluated using 23 well-known benchmark problems and compared with some other meta-heuristic approaches. Considering the benchmark problems, FAWOA achieves best results on 11 problem and only differential evaluation algorithm achieve best results on 10 problems. The rest of the algorithms couldn't achieve the best results on not more than 5 problems. Besides, according to the Friedman and average ranking tests, FAWOA is the first ranked algorithm for solving the benchmark problems. Evaluation results show that the suggested FAWOA approach outperforms the other algorithms as well as the WOA in most of the benchmark problems.

Keywords: fuzzy logic controller, meta-heuristic, whale optimization algorithm, optimization

1.0 INTRODUCTION

Numeric optimization problems are widely investigated among researchers. These problems are the optimization problems that their decision variables are defined in continuously structured solution space, and the aim of solving these problems is optimizing the objective function either minimization or maximization. Gear, pressure vessel, and coil spring design are some real-world engineering problems commonly issued on numeric optimization concept. To solve numerical optimization problems, heuristic algorithms together with meta-heuristic approaches are considered. Heuristic algorithms have been proven to be a comprehensive tool for solving these kinds of problems. Considering that these algorithms are related to problems, the adaptation of the algorithm to optimization problems may become more complex. Metaheuristic algorithms overcome this problem with a growing phenomenon. Because, the performance of the metaheuristic algorithms can clearly be demonstrated on the benchmark problems which have different characteristics (multimodal, unimodal, separable, etc.) and measure abilities (exploration and exploitation capabilities) over the algorithms.

Several meta-heuristic algorithms are presented in the literature. These meta-heuristic algorithms are classified into four major classes depending on their basis, including evolution-based, physics-based, swarm-based, and human-based algorithms. Evolution-based meta-heuristic algorithms are developed by inspiring the laws of natural evolution. Initially, the population is generated stochastically. During the generation process, the best individuals are chosen to produce or affect the next generation until the stopping criteria are reached. The most common algorithms in this concept are genetic algorithms (GA) [1], evolution strategy (ES) [2], and differential evolution (DE) [3]. Physics-based algorithms including simulated annealing (SE) [4], gravitational search algorithm (GSA) [5], and charged system search (CSS) algorithm [6] as the popular ones are developed by simulating the physical rules in the universe. According to this class of algorithms, the population is also generated stochastically, and individuals interact themselves by using physical measures like energy, mass, force, and proximity. The evolutions proceed by modifying these physical measures to get better solutions until the stopping criteria are reached. Swarm-based meta-heuristic algorithms are developed by inspiring the behaviors of the group of animals and interactions between them. In the beginning, the population is generated randomly in the search space, and individuals mimic the movements and interaction of the animals during the generations. Depending upon the best solution of the individual obtained so far or the best solution of the swarm obtained so far or also both of these achievements, the swarm evolves until the stopping criteria are reached. The particle swarm optimization (PSO) algorithm [7] is one of the most investigated meta-heuristic algorithms in this class. There are also many other swarm-based meta-heuristic algorithms exist including ant colony optimization (ACO) algorithm [8], artificial bee colony (ABC) algorithm [9], whale optimization algorithm (WOA) [10], cuckoo search (CS) algorithm [11] and bat algorithm (BA) [12]. The last class of meta-heuristic algorithms, which is based on human, are developed by advancement in the level of searching strategy. The most commonly employed algorithms of this class can be given as tabu search (TS) algorithm [13, 14], harmony search (HS) algorithm [15], and teaching-learning-based optimization (TLBO) algorithm [16].

Among these diverse meta-heuristic algorithms, WOA is one of the new meta-heuristic algorithms used for numeric optimization problems. It has been developed by inspiring the helical movement habits of the humpback whales during hunting. Accordingly, it can be observed in their hunting method that there are three phases, including searching for prey, bubble-net attack, and encircling prey phase. Individuals realize one of these steps stochastically for exploration or exploitation of the search space [10]. There are several studies outperformed on solving optimization problems by using WOA itself, modified WOA versions, and hybrid algorithms based on WOA. By using the WOA, Singh and Prakash outperformed the traditional placement strategies on fiber-wireless network units' placement problem. The authors placed the optical network units by using WOA and compared the results of this placement with the results of the greedy and moth flame optimization algorithms [17]. Azizi et al. presented an upgraded WOA for solving fuzzy logic-based seismic vibration control of a nonlinear steel structure. The authors utilized an upgraded WOA to optimize the parameters of the fuzzy logic controller system which is used for tackling the parameters of calculated control force and the response of the high-rise building structures. The upgraded WOA was constructed by updating the agents in a discrete-time concept that's updating the agents' positions after each iteration ends [18]. Sun et al. utilized quadratic interpolation to improve WOA for solving high-dimensional global optimization problems. The authors modified the update equations of WOA by using quadratic interpolation and evaluate their WOA approach on high dimensional problems [19]. Yin et al. introduced an improved version of WOA in the classification of brain tumours. For WOA to converge faster, chaos theory and logistic mapping algorithm are employed, and the multi-layered perceptron network was used to classify the features of the brain tumour images by using the weights obtained from improved WOA [20]. In another study, Guo et al. proposed an improved version of WOA for forecasting water resource demand. The improved WOA was based on social learning and wavelet mutation approaches for increasing the global search ability of the algorithm and escaping from the local optimums [21]. For spam profile detection on the Twitter network, Krithiga and Ilavarasan presented a study based on the hybridization of WOA with the salp swarm algorithm. In the proposed method, the salp swarm algorithm was implemented on the searching process of WOA, that is by a new parameter, the solutions are updated based on the location of the best food source or the follower's position [22]. Considering the same Twitter spammer detection problem, Al-Zoubi et al. introduced a novel approach for evolving support vector machine as a classifier by utilizing WOA. The authors construct a feature selection mechanism by using the WOA, and implement the classification process on SVM by using this mechanism in each phase. The authors compare their approach with different optimization algorithms as the feature selectors, and different lingual context as Arabic, English, Spanish, and Korean as the context of the inputs [23].

Dealing with the fuzzy logic controller (FLC) based optimization, there are several studies carried out in literature. Lamamra et al. suggested a new approach for controlling the quality of non-linear systems by using a FLC. The authors aimed to optimize three objective functions including a cost function, the number of fuzzy inference rules, and the maximum instantaneous quadratic error. The evaluation showed that the third objective function allowed the improvement of the control quality for complex systems [24]. Solihin et al. proposed a meta-heuristic based optimization approach for tuning the parameters of the FLC which is utilized on optimizing granny cane system. In the study, several meta-heuristic optimization algorithms were evaluated on optimizing the parameters of the FLC system, and PSO was found the best approach on the evaluation [25]. Kaya et al. presented a study for optimization of a digital holographic setup by a fuzzy logic prediction system. During the optimization process, the authors also tried to decrease the required time of the optimization process [26].

Meta-heuristic algorithms are algorithms designed to solve a particular optimization problem without guaranteeing the optimal solution but offer near-optimal solutions in pre-defined time and computational usage. The most challenging mission in the development or improvement of any meta-heuristic algorithm is to find a satisfactory balance in both intensification and diversification [27].

Considering WOA, individuals are updated within one of the three phases during the algorithm. The selection of the phases is implemented by two criteria. The first criterion depends on half of the total iteration cycle and is related to the selection of the exploration or exploitation-based phases. Note that, WOA does not implement the exploration phase after the first half of the total iteration cycle. The second criterion depends on a randomly generated value compared with a pre-defined parameter in the determination of the selection of the phases. In both criteria, the WOA's update procedure involves updating each dimension of individuals. There is no given detail or parameter presented in the research to deal with this issue concerning the effect on the success rate.

FLC is a commonly used method for the determination of the parameters of the algorithms adaptively. In particular, fuzzy logic controllers are used to adaptively adjust the inertia weight parameters in PSO [28-30]. In [31], fuzzy control systems were used to set the balance of exploration and exploitation of the genetic algorithm. In the same way, the teaching learning-based optimization algorithm, Mamdani fuzzy interface was used for exploration and exploitation balance to prevent premature convergence [32]. This process includes presenting some of the variables and the measures of the algorithms as inputs of the FLC model and utilizing the output values adaptively generated by the model as the variables or measures.

To determine which ratio of the dimensions of the individuals in WOA would be updated, a novel FLC-based WOA approach is presented in this study. The suggested approach is evaluated by using 23 well-known benchmark problems, and also the results are compared with the results of 5 different meta-heuristic algorithms, including the original WOA. Besides, Wilcoxon statistical test is performed for the similarity comparison of the algorithms.

The rest of the paper is organized by presenting the original WOA in section 2, introducing the suggested FAWOA approach in section 3 and presenting the experiments and discussion on section 4 and also concluding the study in section 5.

2.0 WHALE OPTIMIZATION ALGORITHM

WOA is a bio-inspired optimization algorithm introduced by Mirjalili et al. in 2016 [10]. The algorithm is modeled by mimicking the hunting strategies of humpback whales and has three main phases, including encircling prey, bubble-net attacking, and search for prey. WOA starts with randomly initializing the candidate solutions in the population and obtaining the fitness value of each candidate solution. After determining the candidate solution with the best fitness value (maximum or minimum concerning the problem), positions of each candidate solution within the population are updated by selecting one of the three phases of the algorithm regarding the parameters $|\vec{A}|$ and p until the stopping criterion or pre-defined maximum number of iterations is reached. Hereby, the parameter p is a random number generated within the range [0,1] and compared with 0.5 for determining the implementation of one of the three main phases. If p is greater than or equal to 0.5 the, candidate solution moves towards the best solution by executing a spiral movement, if not the parameter $|\vec{A}|$ is considered. The parameter $|\vec{A}|$ is linearly decreased from a pre-determined value which is usually 2 to 0 relative to the current iteration, and if $|\vec{A}| < 1$ the candidate solution moves to the current best solution linearly, otherwise, the candidate solution moves to a randomly selected candidate solution linearly. The main phases of WOA are given below and the pseudo-code of the algorithm can be briefly shown as in Figure 1.

2.1 Encircling prey

The algorithm determines the current best solution in this phase. It is assumed that the optimal solution is the current best solution, or it is close to the current best solution. For this reason, the positions of other candidate solutions are updated so that they approach towards the current best solution. The encircling prey phase is represented by Equation (1) and Equation (2).

$$\vec{D} = |\vec{C} \cdot \overrightarrow{X^*}(t) - \vec{X}(t)| \tag{1}$$
$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} \tag{2}$$

where $\vec{X}$ is the position vector (the solution), $\overrightarrow{X^*}$ is the position vector of the current best solution and $\vec{A}$ and $\vec{C}$ are coefficient vectors calculated by using Equation (3) and Equation (4).

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{3}$$
$$\vec{C} = 2\vec{r} \tag{4}$$

where $\vec{a}$ is a vector linearly decreasing from a pre-determined value which is usually 2 to 0 over iterations and $\vec{r}$ is a randomly generated vector within the range [0, 1].

Randomly initialize the agents $\overrightarrow{X_i}(i = 1,2, \dots, n)$
Evaluate the fitness values of each agent
Find the best agent $\overrightarrow{X^*}$
Repeat
    ***For** each agent within the population*
        Update parameters a**,** A**,** C*,* l *and* p
        if *(*$p < 0.5$*)*
            if *(*$|A| < 1$*)*
                Update the positions of the current agent by Eq. 1 & 2
            else if *(*$|A| \geq 1$*)*
                Choose an agent randomly (denoted by X_{rand}*)*
                Update the positions of the current agent by Eq. 8 & 9
            end if
        else if *(*$p \geq 0.5$*)*
                Update the positions of the current agent by Eq. 5 & 6
        end if
    End for
    Check whether an agent violates the boundaries of the search space and fix the positions of the agents if necessary
    Calculate the fitness of each agent
    Update the best agent $\overrightarrow{X^*}$ *if any better solution is found*
    Increase the iteration counter ($t = t + 1$*)*
***Until** the maximum number of iterations*

Figure 1: Pseudo-code of the WOA

2.2 Bubble-net attacking

In the Encircling prey phase, the candidate solutions approximate linearly to the best solution using Equation (2). The bubble-net attacking phase is related to the helix-shaped movement of the Humpback whales since they can also approach their prey by a helix-shaped movement. This helix-shaped motion is mathematically expressed in Equation (5) and Equation (6).

$$\overrightarrow{D'} = |\overrightarrow{X^*}(t) - \vec{X}(t)| \tag{5}$$

$$\vec{X}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) \tag{6}$$

where $\overrightarrow{D'}$ indicates the distance between the solution and the current best solution, b is a constant used for defining the shape of the logarithmic spiral and l is a random number within the range [-1, 1].

Meanwhile, the candidate solutions also have a probability of 50% to make a linear or spiral movement in this phase. Hence, the mathematical expression concerning the movement of the solutions is shown in Equation (7).

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} & if\ p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p \geq 0.5 \end{cases} \tag{7}$$

where p is a random number within the range [0, 1].

2.3 Search for prey

When the random values in vector $\vec{A}$ have values greater than 1 or less than -1, the candidate solutions pursue large steps towards the solutions they refer to. This condition increases the exploration ability of the algorithm. In the search for prey phase, the ability to exploit is increased by using a randomly selected solution instead of using the best solution as a reference. For this reason, Equation (8) and Equation (9) are used to change the positions of the solutions when $|\vec{A}| > 1$.

$$\vec{D} = |\vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X}| \tag{8}$$
$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \tag{9}$$

where $\overrightarrow{X_{rand}}$ is a random position vector chosen among the current population.

3.0 FLC-based WOA

There is always a demand been existing for balancing the exploration and exploitation characteristics of WOA on finding an optimum solution as well as the other meta-heuristic algorithms, and WOA establishes this balance by performing a search for prey (for exploration) and bubble-net attacking (for exploitation) phases. WOA executes the bubble-net attacking phase with a 50% probability and encircling the prey phase or search for prey phase with a 50% probability. Accordingly, the algorithm performs exploitation with a 50% probability. With a 50% probability, the algorithm also performs exploration depending on a random solution on the first half of total iterations and performs exploitation depending on the best solution obtained so far on the second half of the total iterations. Thus, it can be seen that WOA does not execute any exploration over the search space except for the first half of the total iterations. Besides, the trade-off between exploration and exploitation depends only on the selection of the phases regarding the $|\vec{A}|$ parameter.

To find the optimum solution, all dimensions of the individuals are updated simultaneously in WOA, even in exploration or exploitation phases. Modification of all dimensions of the individuals in each iteration cycle may cause a negative effect on the exploitation characteristic of the algorithm. In this study, an FLC-based adaptive update mechanism is suggested to increase the exploitation characteristic of WOA, and hereinafter the suggested approach is referred to as FAWOA. Accordingly, FLC determines the ratio to which dimensions of the candidate solutions are updated in each iteration cycle, and randomly chosen dimensions of the candidate solutions are updated regarding this ratio.

A three input one output Mamdani type FLC is designed in the suggested approach. Current iteration number (*Iteration*), the average of the ten most recent change ratio (*Avg.Chg.Rate*), and standard deviation of the normalized fitness values of the candidate solutions (*N.Std.Dev.*) are used as inputs. The ratio to which dimensions of the candidate solutions are updated (*Chg.Rate*) is the output of the FLC. It should be noted that the main aim of the construction of the input parameters of FLC is to increase the exploitation characteristic of the algorithm. Hereby, the iteration is used as the first parameter to perform more exploration earlier and more exploitation towards the end of the algorithm, which is also used within the WOA. The aim of using the average of the ten most recent change rates is to ensure that the previous changes affect the following change. The standard deviation of the normalized fitness values of the candidate solutions is also utilized to deal with population diversity.

In the suggested approach, FLC determines the ratio to which dimensions of the candidate solutions are updated in each iteration cycle. Then, randomly chosen dimensions of the candidate solutions are updated regarding this ratio, and the number of the dimensions are calculated by using the following equation;

$$rnd_{size} = \begin{cases} \lfloor dim * cr \rfloor & cr \neq 0 \\ 1 & else \end{cases} \tag{10}$$

where rnd_{size} stands for the number of dimensions going to be updated, dim stands for the dimension of the problem and cr stands for change rate obtained by FLC. In an example, if the change rate is obtained as 25% by FLC for a 30-dimensional problem, random selected 7 dimensions of the candidate solution is updated, and rest of the dimensions are used as they are.

To ensure that candidate solutions can exploit sufficiently, to ensure that candidate solutions to be updated will not remain premature, and to prevent early convergence, determination of the dimensions which are going to be updated by using FLC is carried out after a predetermined stage of the total iterations in the suggested approach. In other words, all dimensions of the candidate solutions are updated until a predetermined $I_u{}^{th}$ iteration, and then the FAWOA determines which dimensions are going to be updated to the end of the iterations. The flowchart of the suggested FAWOA approach is given in Figure 2.

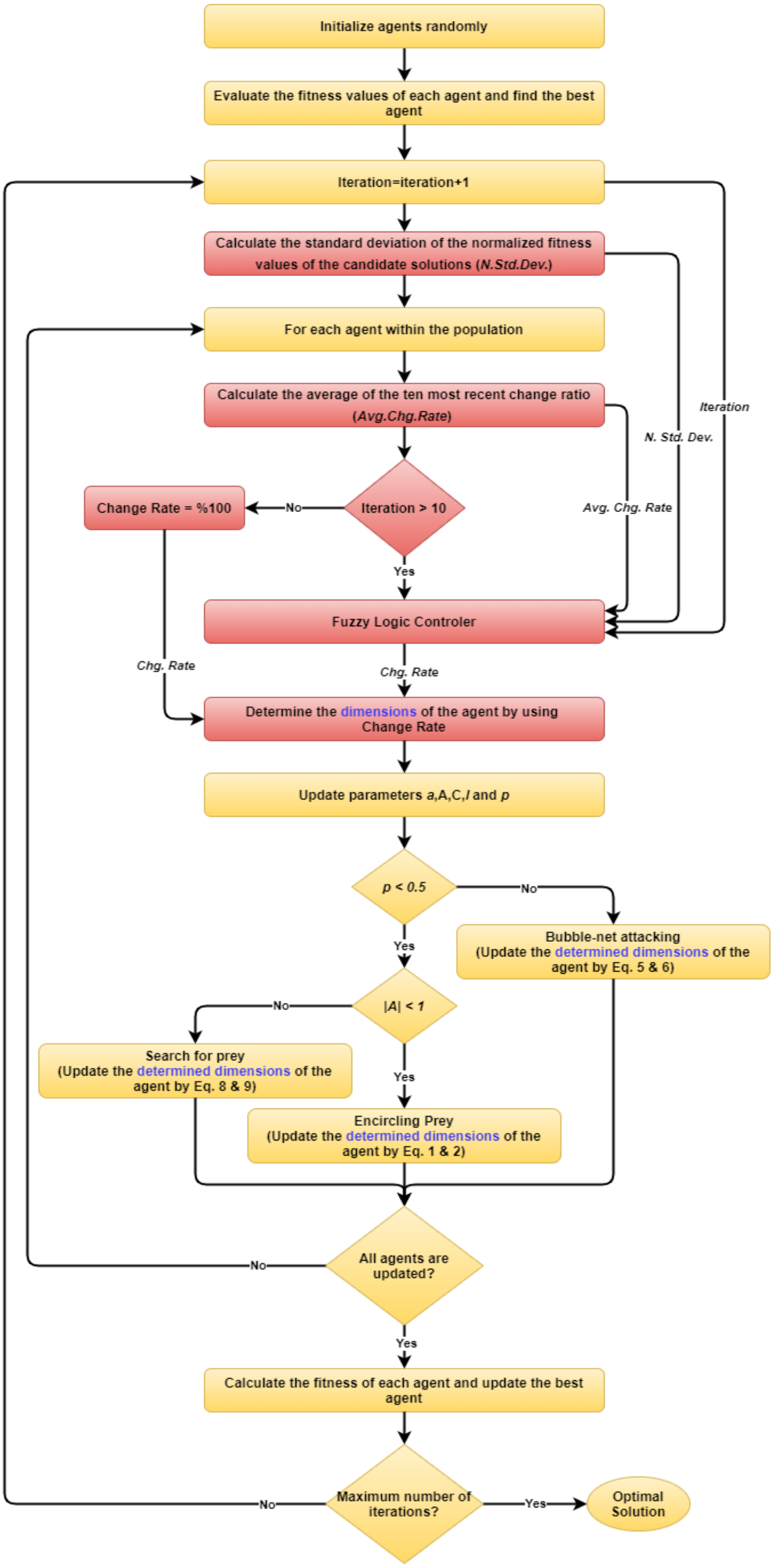

Figure 2: The flowchart of FAWOA

4.0 RESULT & DISCUSSION

The suggested approach FAWOA has been compared with well-known meta-heuristic optimization algorithms including PSO, GSA, DE, and GA, as well as the original version of WOA. The algorithms are evaluated using 23 numeric optimization benchmark problems [33-36] having different characteristics. Dimensions, search ranges, optimum values, and mathematical formulations of these benchmark problems are given in Table 1 and Table 2. Hereby, *F*, *D*, and *C* determine the number of the benchmark function, dimension of the benchmark function, and class of the benchmark function, respectively. Moreover, the benchmark functions are classified into three categories, including unimodal, multimodal, and fixed-dimension multimodal.

Table 1: Benchmark functions (unimodal & multimodal)

F	D	C	Search Range	Optimum Value	Formulation
1	30	U	[−100, 100]	0	$F_1(x) = \sum_{i=1}^{n} x_i^2$
2	30	U	[−10, 10]	0	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
3	30	U	[−100, 100]	0	$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$
4	30	U	[−100, 100]	0	$F_4(x) = \max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$
5	30	U	[−30, 30]	0	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
6	30	U	[−100, 100]	0	$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$
7	30	U	[−1.28, 1.28]	0	$F_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$
8	30	M	[−500, 500]	-418.9829x5	$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$
9	30	M	[−5.12, 5.12]	0	$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$
10	30	M	[−32, 32]	0	$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$
11	30	M	[−600, 600]	0	$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$
12	30	M	[−50, 50]	0	$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$
13	30	M	[−50, 50]	0	$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$

Three triangle membership functions are constructed as small (*S*), medium (*M*) and large (*L*) for each input and output parameter in the FLC. These membership functions are given in Figure 3. The rule set is constructed including 27 fuzzy rules which are given in Table 3, and mean and maximum methods are employed as defuzzification.

Table 2: Benchmark functions (fixed-dimension multimodal)

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

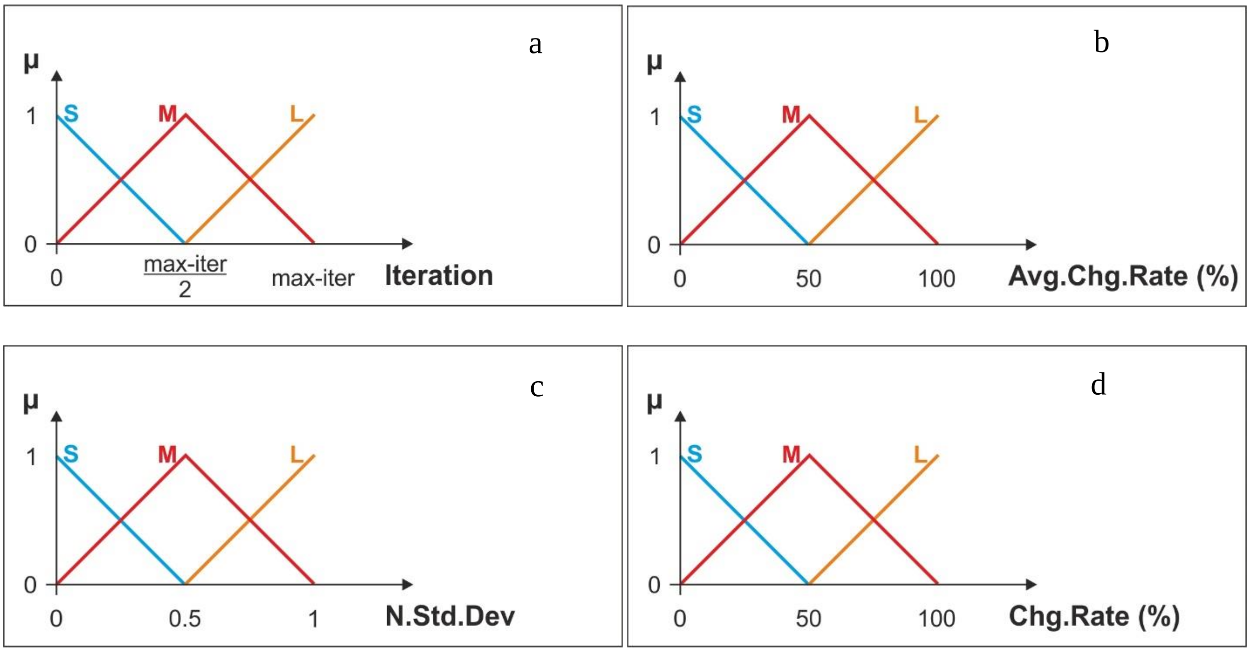

Figure 3: Membership functions of inputs (a, b and c) and output (d)

Table 3: Fuzzy rules of the fuzzy logic controller

Rule No	**Input**			**Output**
	Iteration	**Avg.Chg.Rate (%)**	**N.Std.Dev**	**Chg.Rate(%)**
1	S	S	S	L
2	S	S	M	L
3	S	S	L	L
4	S	M	S	L
5	S	M	M	L
6	S	M	L	L
7	S	L	S	L
8	S	L	M	L
9	S	L	L	L
10	M	S	S	L
11	M	S	M	M
12	M	S	L	M
13	M	M	S	L
14	M	M	M	L
15	M	M	L	M
16	M	L	S	S
17	M	L	M	S
18	M	L	L	S
19	L	S	S	M
20	L	S	M	M
21	L	S	L	M
22	L	M	S	S
23	L	M	M	S
24	L	M	L	S
25	L	L	S	L
26	L	L	M	M
27	L	L	L	S

Each algorithm is run 30 times with different random seeds, and the results are presented as the average of the results of these runs. Besides, the population size and the maximum fitness evaluation are used as being equal to 30 and 15,000 in each experiment, respectively. The comparative results are given in Table 4 and Table 5, including the mean results and standard deviations, indicating the best results in bold font face for each benchmark problem. The parameter setup of each algorithm, including PSO, GSA, DE, GA, and FAWOA, is done by using the same values within the original study of WOA [10]. Additionally, the I_u is used as 0.2 since both WOA and FAWOA are tend to exploit when the iteration is greater than half (0.5) of the total iterations.

In order to demonstrate if there is a significant difference between FAWOA and each other algorithm exist, the Wilcoxon test is applied. The results of the Wilcoxon test are given in Table 4 and Table 5. Hereby, the Wilcoxon test is evaluated between FAWOA and each other algorithm with a confidence value of 0.05. The (+) sign states that a significant difference exists statistically between the algorithms (p-value < 0.05), and the (-) sign states the opposite (p-value ≥ 0.05). Friedman's test and average ranking test are also applied for evaluating the algorithms with a confidence value of 0.05, and results of the Friedman test and average ranking of the algorithms are given in Table 4 and Table 5.

Table 4: Comparison of results of the benchmark functions F1-F13

Functions		FAWOA	WOA	PSO	GSA	DE	GA
F1	*Mean*	9.5397E-45	**4.0083E-73**	6.7824E-06	2.5208E-16	4.2960E-04	5.2221E+01
	Std. Dev.	4.4368E-44	**2.1569E-72**	2.3558E-05	1.4509E-16	2.1191E-04	1.9500E+01
	Rank / Sign	2 / (NA)	**1 / (+)**	4 / (+)	3 / (+)	5 / (+)	6 / (+)
F2	*Mean*	2.1875E-29	**1.3870E-51**	2.7731E-02	5.9073E-02	2.1903E-03	1.9675E+00
	Std. Dev.	5.2401E-29	**3.6744E-51**	7.5027E-02	3.0541E-01	3.9752E-04	5.4514E-01
	Rank / Sign	2 / (NA)	**1 / (+)**	4 / (+)	5 / (+)	3 / (+)	6 / (+)
F3	*Mean*	3.8812E+03	4.5335E+04	**5.5916E+02**	1.0193E+03	3.1878E+04	5.6013E+03
	Std. Dev.	2.0544E+03	1.3247E+04	**1.8113E+02**	4.5441E+02	5.0895E+03	1.7230E+03
	Rank / Sign	3 / (NA)	6 / (+)	**1 / (+)**	2 / (+)	5 / (+)	4 / (+)
F4	*Mean*	**4.1709E+00**	4.3985E+01	4.8962E+00	7.7193E+00	1.2906E+01	1.0127E+01
	Std. Dev.	**2.7920E+00**	2.7938E+01	8.9269E-01	1.8473E+00	1.7543E+00	1.4941E+00
	Rank / Sign	**1 / (NA)**	6 / (+)	2 / (+)	3 / (+)	5 / (+)	4 / (+)
F5	*Mean*	**2.6114E+01**	2.7812E+01	4.8680E+01	8.1713E+01	1.6144E+02	1.7920E+03
	Std. Dev.	**9.3318E-01**	4.2148E-01	3.2744E+01	9.9094E+01	4.7577E+01	7.2394E+02
	Rank / Sign	**1 / (NA)**	2 / (+)	3 / (+)	4 / (+)	5 / (+)	6 / (+)
F6	*Mean*	4.1256E-02	4.4007E-01	**5.2414E-06**	2.5843E-04	4.5898E-04	4.9868E+01
	Std. Dev.	1.1050E-01	1.5448E-01	**1.4774E-05**	1.3917E-03	1.9941E-04	2.2747E+01
	Rank / Sign	4 / (NA)	5 / (+)	**1 / (+)**	2 / (+)	3 / (+)	6 / (+)
F7	*Mean*	**3.7090E-03**	4.0131E-03	2.6123E-02	8.6413E-02	5.9135E-02	6.7321E-02
	Std. Dev.	**2.8870E-03**	3.4676E-03	8.4258E-03	3.5304E-02	1.4308E-02	2.1470E-02
	Rank / Sign	**1 / (NA)**	2 / (+)	3 / (+)	5 / (+)	4 / (+)	6 / (+)
F8	*Mean*	**-1.0486E+04**	-9.8039E+03	-6.8620E+03	-2.6751E+03	-9.8647E+03	-1.0190E+04
	Std. Dev.	**1.6164E+03**	1.9772E+03	7.5401E+02	3.2923E+02	5.7432E+02	3.9991E+02
	Rank / Sign	**1 / (NA)**	4 / (+)	5 / (+)	6 / (+)	3 / (+)	2 / (+)
F9	*Mean*	4.7031E+00	**5.6843E-15**	4.9781E+01	3.0180E+01	8.7946E+01	3.4718E+01
	Std. Dev.	1.9509E+01	**2.2499E-14**	1.1636E+01	7.6092E+00	8.0015E+00	7.0595E+00
	Rank / Sign	2 / (NA)	**1 / (+)**	5 / (+)	3 / (+)	6 / (+)	4 / (+)
F10	*Mean*	**4.5593E-15**	5.1514E-15	1.1167E-01	6.2087E-02	5.5309E-03	2.7854E+00
	Std. Dev.	**2.3357E-15**	3.2302E-15	3.3144E-01	2.3231E-01	1.6585E-03	3.5838E-01
	Rank / Sign	**1 / (NA)**	2 / (+)	5 / (+)	4 / (+)	3 / (+)	6 / (+)
F11	*Mean*	**0.0000E+00**	1.0823E-02	1.3438E-02	2.8003E+01	1.0561E-02	1.5022E+00
	Std. Dev.	**0.0000E+00**	4.0535E-02	1.6551E-02	5.8603E+00	1.2711E-02	2.4100E-01
	Rank / Sign	**1 / (NA)**	4 / (+)	3 / (+)	6 / (+)	2 / (+)	5 / (+)
F12	*Mean*	6.0318E-03	2.3283E-02	7.2591E-02	1.8547E+00	**4.8592E-05**	2.5866E-01
	Std. Dev.	6.1761E-03	9.9633E-03	1.5650E-01	9.7225E-01	**2.7605E-05**	1.6017E-01
	Rank / Sign	2 / (NA)	3 / (+)	4 / (+)	6 / (+)	**1 / (+)**	5 / (+)
F13	*Mean*	1.0561E-01	4.5608E-01	6.9696E-03	7.7333E+00	**2.3494E-04**	2.2716E+00
	Std. Dev.	2.1532E-01	2.2076E-01	1.1161E-02	5.8785E+00	**1.1850E-04**	9.8764E-01
	Rank / Sign	3 / (NA)	4 / (+)	2 / (+)	6 / (+)	**1 / (+)**	5 / (+)
		FAWOA	**WOA**	**PSO**	**GSA**	**DE**	**GA**

For demonstrating the convergence characteristic of the suggested FAWOA, the convergence graphs of the selected benchmark functions including F1, F2, F4, F6, F9, F11, F13, and F15 are given in Figure 4.

Table 5: Comparison of results of the benchmark functions F14-F23

Functions		FAWOA	WOA	PSO	GSA	DE	GA
F14	*Mean*	2.3713E+00	2.7679E+00	3.9541E+00	6.0193E+00	**9.9800E-01**	**9.9800E-01**
	Std. Dev.	2.9047E+00	2.8473E+00	2.6376E+00	4.1889E+00	**6.6613E-16**	**6.6613E-16**
	Rank / Sign	3 / (NA)	4 / (+)	5 / (+)	6 / (+)	**1.5 / (+)**	**1.5 / (+)**
F15	*Mean*	**5.6428E-04**	9.3590E-04	1.3182E-03	4.7222E-03	7.3587E-04	4.4060E-03
	Std. Dev.	**3.8176E-04**	2.1436E-03	3.5655E-03	4.0332E-03	4.9189E-05	6.4532E-03
	Rank / Sign	**1 / (NA)**	3 / (+)	4 / (+)	6 / (+)	2 / (+)	5 / (+)
F16	*Mean*	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**
	Std. Dev.	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Rank / Sign	**3.5 / (NA)**	**3.5 / (-)**	**3.5 / (-)**	**3.5 / (-)**	**3.5 / (-)**	**3.5 / (-)**
F17	*Mean*	**3.9789E-01**	**3.9789E-01**	**3.9789E-01**	**3.9789E-01**	**3.9789E-01**	3.9790E-01
	Std. Dev.	**1.6653E-16**	**1.6432E-05**	**1.6653E-16**	**1.6653E-16**	**1.6653E-16**	2.6391E-05
	Rank / Sign	**3 / (NA)**	**3 / (-)**	**3 / (-)**	**3 / (-)**	**3 / (-)**	6 / (+)
F18	*Mean*	**3.0000E+00**	3.0001E+00	**3.0000E+00**	**3.0000E+00**	**3.0000E+00**	5.7007E+00
	Std. Dev.	**0.0000E+00**	1.8167E-04	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	8.1022E+00
	Rank / Sign	**2.5 / (NA)**	5 / (+)	**2.5 / (-)**	**2.5 / (-)**	**2.5 / (-)**	6 / (+)
F19	*Mean*	**-3.8623E+00**	-3.8549E+00	-3.8370E+00	-3.8628E+00	-3.8628E+00	-3.8628E+00
	Std. Dev.	**1.9656E-03**	1.4160E-02	1.3876E-01	4.4409E-16	4.4409E-16	4.4409E-16
	Rank / Sign	**1 / (NA)**	5 / (+)	6 / (+)	3 / (+)	3 / (+)	3 / (+)
F20	*Mean*	-3.2598E+00	-3.1616E+00	-3.2824E+00	**-3.3220E+00**	**-3.3220E+00**	-3.2863E+00
	Std. Dev.	7.5345E-02	1.4949E-01	5.6050E-02	**1.7764E-15**	**1.7764E-15**	5.4484E-02
	Rank / Sign	5 / (NA)	6 / (+)	4 / (+)	**1.5 / (+)**	**1.5 / (+)**	3 / (+)
F21	*Mean*	-8.0331E+00	-7.5505E+00	-6.1373E+00	-7.9419E+00	**-9.7423E+00**	-6.0672E+00
	Std. Dev.	2.6308E+00	3.0913E+00	3.2044E+00	3.3813E+00	**1.2460E+00**	3.4403E+00
	Rank / Sign	2 / (NA)	4 / (+)	5 / (+)	3 / (+)	**1 / (+)**	6 / (+)
F22	*Mean*	-7.5595E+00	-7.6307E+00	-7.3594E+00	-1.0069E+01	**-1.0357E+01**	-5.3241E+00
	Std. Dev.	2.9235E+00	3.0164E+00	3.5410E+00	1.2551E+00	**1.6363E-01**	3.3647E+00
	Rank / Sign	4 / (NA)	3 / (+)	5 / (+)	2 / (+)	**1 / (+)**	6 / (+)
F23	*Mean*	-7.2914E+00	-8.1565E+00	-6.6473E+00	-1.0457E+01	**-1.0536E+01**	-7.7603E+00
	Std. Dev.	2.9125E+00	3.2999E+00	3.9221E+00	4.2647E-01	**8.8818E-15**	3.6643E+00
	Rank / Sign	5 / (NA)	3 / (+)	6 / (+)	2 / (+)	**1 / (+)**	4 / (+)
		FAWOA	**WOA**	**PSO**	**GSA**	**DE**	**GA**
Average Ranking Results							
	Overall Rank	54	80.5	86	87.5	66	109
	Mean Rank	2.35	3.50	3.74	3.80	2.87	4.74
	Final Rank	1	3	4	5	2	6
Friedman Test Results							
	Mean Rank	2.48	3.46	3.78	3.80	2.83	4.65
	Final Rank	1	3	4	5	2	6

Considering the test results, it can be seen that the FAWOA achieves better or equal results compared to the rest of the algorithms for 11 benchmark functions. Moreover, it can be seen from the convergence graphs that the suggested FAWOA does not stack in terms of convergence in almost every test case. Besides, being an improved modification of WOA, FAWOA has better results and convergence characteristics compared to WOA.

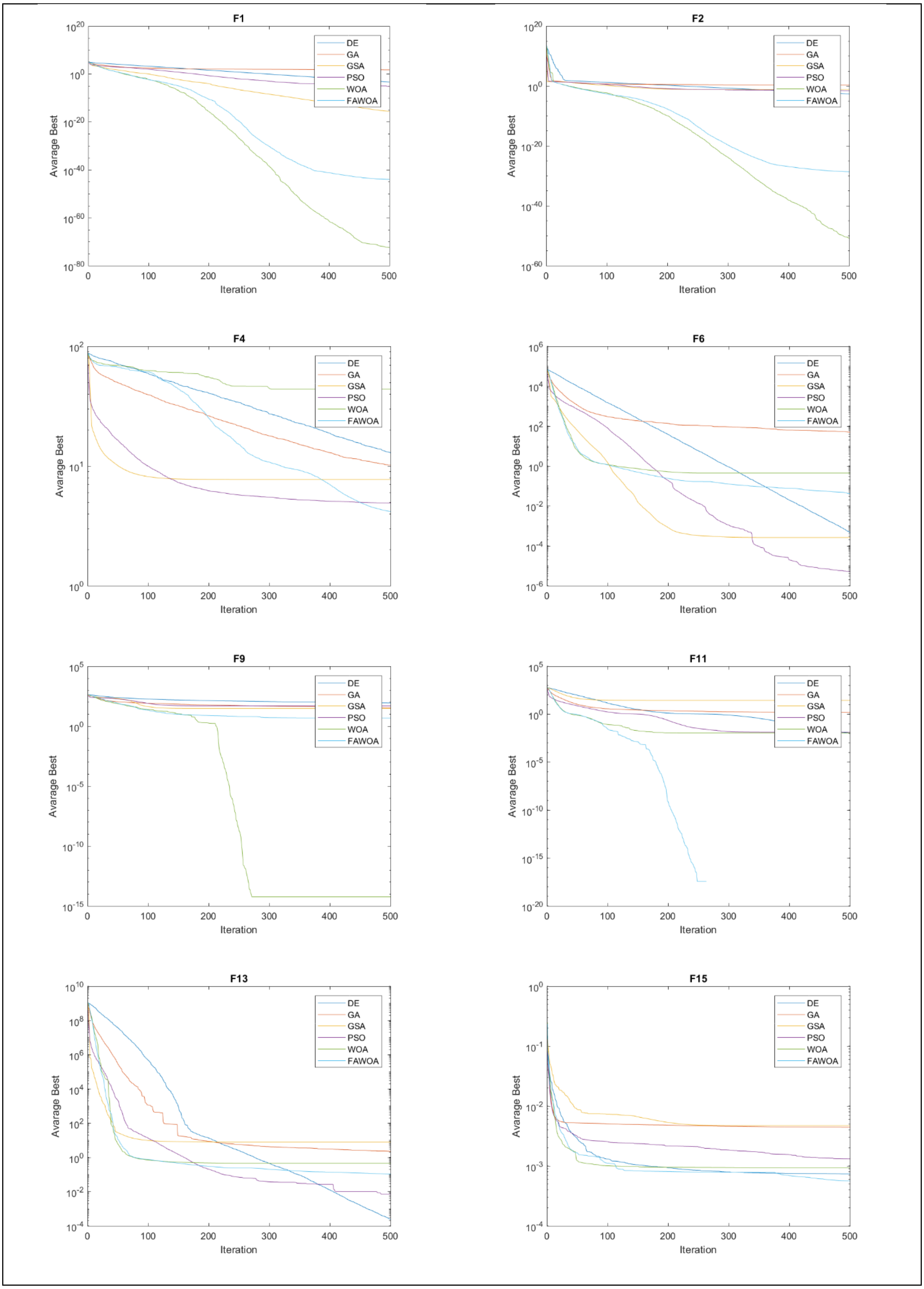

Figure 4: Convergence graphs of functions F1, F2, F4, F6, F9, F11, F13 and F15.

Wilcoxon test demonstrates that FAWOA differs from the rest of the algorithms in comparison to almost in every test case. The FAWOA is demonstrated as not significantly different from the other algorithms in only F16, F17, and F18 benchmark functions. Almost every algorithm in comparison acquires the same results, which are the optimum results for these benchmark functions. The p-value of the Friedman test is obtained as 6.1790E-04. Since the p-value is smaller than the confidence value, it can be said that there is a significant difference between the results of the algorithms. Furthermore, the Friedman ranking results and average ranking results indicate that the suggested FAWOA outperforms the rest of the compared algorithms in usual. Figure 5 demonstrates the mean values of Friedman and the Average ranking results of the evaluation. This figure also reveals the success of the suggested method.

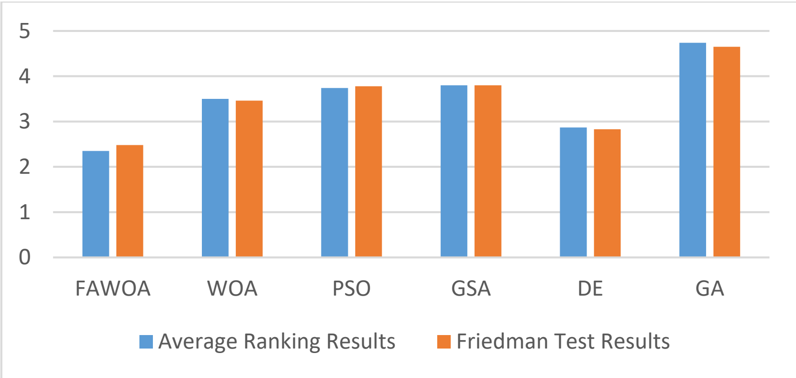

Figure 5: The overall performance results

5.0 CONCLUSION

To improve the exploitation behavior of WOA, a fuzzy logic controller based adaptive WOA is suggested in this study. An FLC realizes the update scheme of WOA, and the proposed FLC determines the rate of the change in terms of dimension. Current iteration number, the average of the ten most recent change ratio, and the standard deviation of the normalized fitness values of the candidate solutions are used as inputs of the FLC. By using the rate obtained by FLC, randomly selected dimensions of the individuals are updated after a predetermined stage of total iterations. The results of the suggested FAWOA are compared with the results of the well-known optimization algorithms, including PSO, GSA, DE, and GA, as well as WOA. The performances of the algorithms are evaluated on 23 benchmark functions. Wilcoxon, Friedman, and average ranking tests are also implemented for the experiments. According to the test results, the suggested FAWOA achieves successful results compared to the other algorithms in comparison. According to the statistical tests, it's observed that the FAWOA significantly differs from the other algorithms, and has the 1st rank for the employed benchmark functions. It's better to note that the suggested FAWOA outperforms the WOA in terms of success rate and convergence.

REFERENCES

[1] J. H. Holland, "Genetic Algorithms," (in English), *Scientific American*, vol. 267, no. 1, pp. 66-72, Jul 1992.

[2] I. Rechenberg, "Evolutionsstrategien," in *Simulationsmethoden in der Medizin und Biologie*, Berlin, Heidelberg, 1978, pp. 83-114: Springer Berlin Heidelberg.

[3] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997/12/01 1997.

[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," (in English), *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[5] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," (in English), *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, Jun 13 2009.

[6] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," (in English), *Acta Mechanica,* vol. 213, no. 3-4, pp. 267-289, Sep 2010.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," (in English), *1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6,* pp. 1942-1948, 1995.

[8] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," (in English), *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics,* vol. 26, no. 1, pp. 29-41, Feb 1996.

[9] D. Karaboğa, "An idea based on honey bee swarm for numerical optimization," in "Technical Report-TR06," Erciyes University, Engineering Faculty, Comput. Eng.Dep.2005.

[10] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," (in English), *Advances in Engineering Software,* vol. 95, pp. 51-67, May 2016.

[11] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *International symposium on stochastic algorithms*, 2009, pp. 169-178: Springer.

[12] X. S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," (in English), *Engineering Computations,* vol. 29, no. 5-6, pp. 464-483, 2012.

[13] F. Glover, "Tabu Search—Part I," *ORSA Journal on Computing,* vol. 1, no. 3, pp. 190-206, 1989.

[14] F. Glover, "Tabu Search—Part II," *ORSA Journal on Computing,* vol. 2, no. 1, pp. 4-32, 1990.

[15] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," (in English), *Simulation,* vol. 76, no. 2, pp. 60-68, Feb 2001.

[16] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems," (in English), *Information Sciences,* vol. 183, no. 1, pp. 1-15, Jan 15 2012.

[17] P. Singh and S. Prakash, "Optical network unit placement in Fiber-Wireless (FiWi) access network by Whale Optimization Algorithm," (in English), *Optical Fiber Technology,* vol. 52, Nov 2019.

[18] M. Azizi, R. G. Ejlali, S. A. M. Ghasemi, and S. Talatahari, "Upgraded Whale Optimization Algorithm for fuzzy logic based vibration control of nonlinear steel structure," (in English), *Engineering Structures,* vol. 192, pp. 53-70, Aug 1 2019.

[19] Y. J. Sun, T. Yang, and Z. J. Liu, "A whale optimization algorithm based on quadratic interpolation for high-dimensional global optimization problems," (in English), *Applied Soft Computing,* vol. 85, Dec 2019.

[20] B. Yin, C. Wang, and F. Abza, "New brain tumor classification method based on an improved version of whale optimization algorithm," (in English), *Biomedical Signal Processing and Control,* vol. 56, Feb 2020.

[21] W. Y. Guo, T. Liu, F. Dai, and P. Xu, "An improved whale optimization algorithm for forecasting water resources demand," (in English), *Applied Soft Computing,* vol. 86, Jan 2020.

[22] R. Krithiga and E. Ilavarasan, "A Reliable Modified Whale Optimization Algorithm based Approach for Feature Selection to Classify Twitter Spam Profiles," *Microprocessors and Microsystems,* p. 103451, 2020/11/16/ 2020.

[23] A. M. Al-Zoubi, H. Faris, J. Alqatawna, and M. A. Hassonah, "Evolving Support Vector Machines using Whale Optimization Algorithm for spam profiles detection on online social networks in different lingual contexts," (in English), *Knowledge-Based Systems,* vol. 153, pp. 91-104, Aug 1 2018.

[24] K. Lamamra, F. Batat, and F. Mokhtari, "A new technique with improved control quality of nonlinear systems using an optimized fuzzy logic controller," *Expert Systems with Applications,* vol. 145, p. 113148, 2020/05/01/ 2020.

[25] M. I. Solihin, C. Y. Chuan, and W. Astuti, "Optimization of fuzzy logic controller parameters using modern meta-heuristic algorithm for gantry crane system (GCS)," *Materials Today: Proceedings,* vol. 29, pp. 168-172, 2020/01/01/ 2020.

[26] G. U. Kaya, O. Erkaymaz, and Z. Sarac, "Optimization of digital holographic setup by a fuzzy logic prediction system," *Expert Systems with Applications,* vol. 56, pp. 177-185, 2016/09/01/ 2016.

[27] R. Rajakumar, P. Dhavachelvan, and T. Vengattaraman, "A survey on nature inspired meta-heuristic algorithms with its domain specifications," in *2016 International Conference on Communication and Electronics Systems (ICCES)*, 2016, pp. 1-6.

[28] S. Yuhui and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001, vol. 1, pp. 101-106 vol. 1.

[29] L. Hongbo and A. Abraham, "Fuzzy adaptive turbulent particle swarm optimization," in *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, 2005, p. 6 pp.

[30] D. Tian and N. Li, "Fuzzy Particle Swarm Optimization Algorithm," in *2009 International Joint Conference on Artificial Intelligence*, 2009, pp. 263-267.

[31] M. Vannucci, V. Colla, and S. Dettori, "Fuzzy Adaptive Genetic Algorithm for Improving the Solution of Industrial Optimization Problems," *IFAC-PapersOnLine,* vol. 49, no. 12, pp. 1128-1133, 2016/01/01/ 2016.

[32] K. Z. Zamli, F. Din, S. Baharom, and B. S. Ahmed, "Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites," *Engineering Applications of Artificial Intelligence,* vol. 59, pp. 35-50, 2017/03/01/ 2017.

[33] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," (in English), *Ieee Transactions on Evolutionary Computation,* vol. 3, no. 2, pp. 82-102, Jul 1999.

[34] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," (in English), *International Journal of Computer Mathematics,* vol. 77, no. 4, pp. 481-506, 2001.

[35] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," (in English), *Swarm and Evolutionary Computation,* vol. 1, no. 1, pp. 3-18, Mar 2011.

[36] W. F. Gao, S. Y. Liu, and L. L. Huang, "A Novel Artificial Bee Colony Algorithm Based on Modified Search Equation and Orthogonal Learning," (in English), *Ieee Transactions on Cybernetics,* vol. 43, no. 3, pp. 1011-1024, Jun 2013.