ISSN 2007-9737

Comparative Study of Gorilla Troops Optimizer and Stochastic Fractal Search with Fuzzy Dynamic Parameter Adaptation

Marylu L. Lagunes*, Ivette Miramontes, Oscar Castillo, Patricia Melin

Tijuana Institute of Technology, TecNM,
Mexico

marylu.lara@tectijuana.edu.mx, cynthiaivette84@gmail.com, {ocastillo, pmelin}@tectijuana.mx

Abstract. Metaheuristics has a very important role today in solving optimization problems; the vast majority of these methods are based on principles that imitate natural processes to achieve their results. The objective of this research is the analysis of the adaptability and stability of two bio-inspired methods, proposing a comparison between two optimization algorithms to evaluate and compare the performance and effectiveness of the algorithms in different optimization problems, the first, inspired by the social behavior of gorillas, which is called Artificial Gorilla Troops Optimizer (GTO), which is mathematically formulated to achieve exploration and exploitation in a given search space. The second algorithm is the one nature-inspired by imitating fractal behavior, known as Stochastic Fractal Search (SFS), where each of the particles moves stochastically until the objective function is found. By comparing both methods using benchmark functions, in this case CEC'2017 functions and performing the corresponding statistical analysis, we can conclude that with the GTO method, we obtained better results, since they are closer to the global optimum of the functions in comparison. with the SFS algorithm.

Keywords. Bio-inspired algorithms, fuzzy logic, optimization, CEC'2017 benchmark functions.

1 Introduction

Bio-inspired algorithms are those methods that try to imitate the behavior of the natural evolution of species in the search for mates and food. These forms of behavior are adapted to provide solutions to real problems in different algorithms.

Among the widely recognized methods are evolutionary algorithms, such as genetic algorithms, which are based on the evolution of species. This method has been considered to solve an endless number of optimization problems as described by the following authors [1, 2, 3, 4, 5, 6, 7]. Its versatility is evident in different contexts, such as in the development of computational algorithms capable of solving various problems in different fields, including pattern recognition, among others.

In the soft computing area, they are used to adjust membership functions in fuzzy controllers which help improve the performance of said systems. Bio-inspired optimization algorithms are characterized by being adaptive and non-deterministic, of which we find different applications today, for example, the design of communications antennas, and military tactics in airplanes [8, 9].

There is a wide variety of bio-inspired algorithms used for optimization, we will start by mentioning the Firefly Algorithm (FA) algorithm which is based on the inspiration of the behavior of flickering fireflies where each of the fireflies emits light or glow to find a partner and food.

This method is governed by 3 main rules and is widely used in the area of soft computing, in the following cases[10, 11, 12, 13, 14]. It describes how fireflies generate different values of possible solutions, for the optimization of fuzzy controllers that manage the behavior of an autonomous robot, in the area of artificial neural networks we can also find the use of bio-inspired algorithms as described in [15, 16, 17, 18, 19, 20], in addition to the mentioned methods, currently there are a wide variety of bio-inspired algorithms for optimization, the choice of them will depend on the problem to be solved, here we cite some[21, 22, 23, 24, 25].

The main objective of this research lies in evaluating and contrasting the performance of two algorithms, the bio-inspired Artificial Gorilla Troops

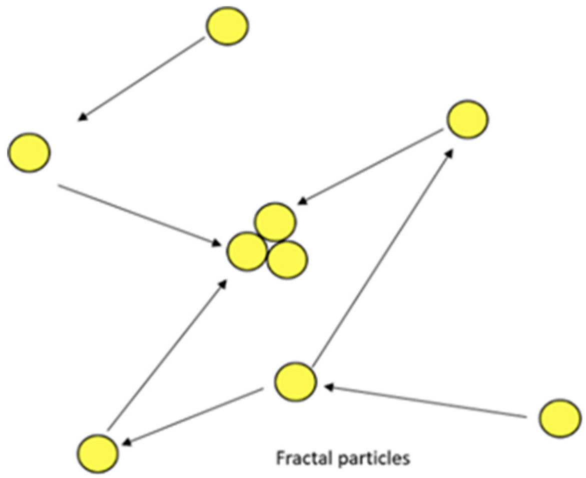

Fig. 1. Fractal particles movement

Optimizer (GTO) [26] and nature-inspired Stochastic Fractal Search (SFS) [27] using mathematical functions CEC 2017 [28].

The aim is to determine its efficiency, adaptability, stability, and effectiveness in various optimization problems. This comparison between algorithms intends to identify which of them is most effective in solving specific problems.

By analyzing their performance in specific cases, the aim is to discover the strengths and weaknesses of each algorithm, which would allow them to be adjusted and improve their applicability in more complex or specialized problems.

This comparative evaluation also contributes to a better understanding of algorithm approaches and their applicability in different problem domains. It is motivated by the search for the optimal solution in specific problems, as well as the dynamic adaptation capacity of the method, through the application of fuzzy logic to various variables to find the best solution.

The adaptability analysis was carried out by applying a fuzzy system, which made it possible to dynamically adjust key variables in each method and observe their behavior in the optimization of mathematical functions, particularly those of CEC 2017.

Fuzzy logic offers a mathematical framework which models non-linear functions, transforming inputs into outputs based on approximate reasoning. In this context, it was used to dynamically adjust the parameters of the compared methods, selecting said variables based on the design of the method and the needs posed by the problem in question.

This article is distributed as follows: Section 2 provides a review of the literature on nature-inspired metaheuristic algorithms used in this comparison and fuzzy logic. Section 3 describes the development of the comparison using fuzzy logic applied to CEC 2017 mathematical functions.

Section 4 observes the results obtained by the algorithms, showing the performance in tables and graphs. Section 5 deals with the discussion where the different points of view are presented according to the results obtained and Section 6 describes the conclusions and future work.

2 Literature Review

In this section, the bio-inspired approaches used in the research are detailed, along with the context related to fuzzy logic.

2.1 Artificial Gorilla Troops Optimizer (Algorithm 1)

Artificial Gorilla Troops Optimizer (GTO) [29] utilizes gorilla social behavior to enhance practical task performance.

It operates on a gorilla squad, where a primary model train specialized models for diverse tasks. These models benefit from knowledge transfer, expediting learning and enhancing outcomes.

The GTO method has demonstrated effectiveness in areas like natural language processing and computer vision, particularly in related responsibilities of varying complexity. Transferring knowledge from complex to specific tasks can enhance overall system performance.

However, it is crucial to note that GTO implementation may vary based on context and tasks, necessitating adjustments according to specific requirements and data [26, 30].

The exploratory phase initiates when a gorilla migrates, considering:

- When migrates to an unknown location.
- When migrates to a known location.
- Moving towards other gorillas.

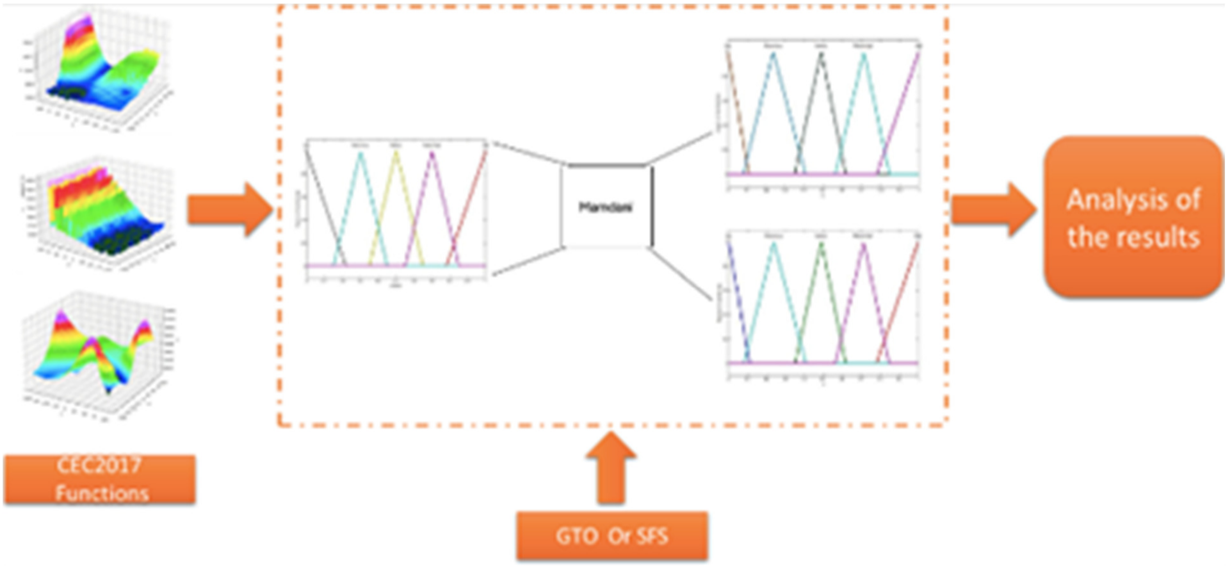

Fig. 2. Graphic description of the optimization process

The parameter P determines the migration mechanism to an unknown location. The initial mechanism activates when Rand < P. Contrariwise, if Rand ≥ 0.5, the movement shifts towards other gorillas. If Rand < 0.5, migration to a known location is favored. Eq. (1) summaries these three mechanisms of the exploratory phase:

$$GX(t+1) = \begin{cases} (UB - LB) \times r_1 + LB & rand < p \\ (r_1 - C) \times X_r(t) + L \times H & rand \geq 0.5 \\ X\,(i) - L \times \left(L \times \left(X(t) - GX_r(t)\right), + r_3 \times \left(X(t) - GX_r(t)\right)\right) & rand < p \end{cases} \quad (1)$$

where, $GX(t+1)$ denotes the individual next iteration candidate position vector at time (t), and $X(t)$ represents the current position vector. The values r_1, r_2, $r_3$and rand are random and range from 0 to 1, updated in each iteration.

The parameter p requires a predefined value before optimization, ranging from 0 to 1, determining the probability of selecting the migration mechanism to an unknown location. UB and LB indicate the upper and lower variable limits, respectively. X_r is randomly chosen from the gorilla population, while GX_r represents one of the candidate position vectors randomly selected, encompassing updated positions in each phase.

Finally, H is computed using equations (2), (4), and (5):

$$\mathrm{C} = \mathrm{F} * \left(1 - \frac{\mathrm{It}}{\mathrm{MaxIt}}\right), \quad (2)$$

$$F = \cos(2 \times r_4) + 1, \quad (3)$$

$$\mathrm{L} = \mathrm{C} \times \mathrm{l}. \quad (4)$$

In (2), t represents the current iteration value, $MaxIt$ stands for the total number of iterations for optimization, and F is determined by (3), wherein the cosine function cos and random values r_4 between 0 and 1 are employed, updating each iteration.

As per (2), initial optimization stages generate values with wide-ranging abrupt changes, which progressively narrow down towards the end.

In (4) computes L, with l being a random value ranging from -1 to 1, simulating the leadership role of the silverback gorilla. While the silverback gorilla may initially lack experience in making optimal decisions for food and group control, it gains stability and expertise over time.

The alterations in values are generated by (2) and (4). Additionally, in (1) calculates H using (5), and this, Z is determined through (6), where Z represents a random value within the problem is dimensions, ranging from -C to C:

ISSN 2007-9737

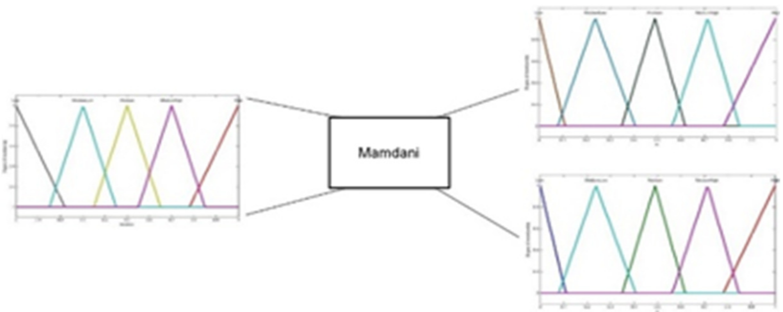

Fig. 3. GTO Triangular fuzzy inference system

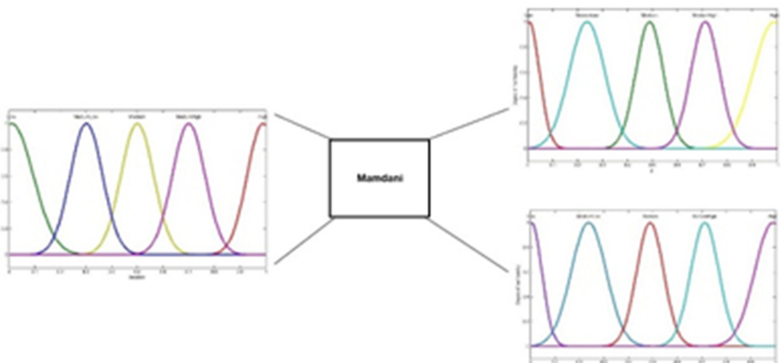

Fig. 4. GTO Gaussian fuzzy Inference system

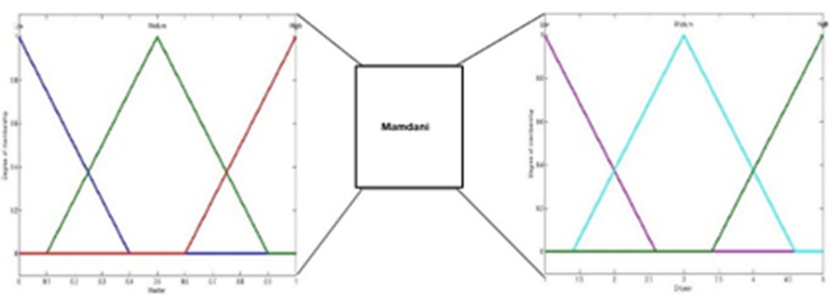

Fig. 5. SFS Triangular fuzzy inference system

$$H = Z \times X(t), \tag{5}$$

$$Z = [-C, C]. \tag{6}$$

In the exploitation phase of the method, two behaviors are employed: "Following the silverback" and "Competition for adult females."

The silverback gorilla leads the troop, makes decisions, leads movements and guides the gorillas to food sources, being responsible for the safety and well-being of the group.

All members adhere to its decisions. Nonetheless, the silverback gorilla can weaken, age, and eventually die, allowing the blackback of the group to assume leadership or other male gorillas to confront and dominate the silverback.

The decision between "Follow the Silverback" and "Competition for adult females" is made by the C value in (2). If C ≥ W, "Follow the silverback" is chosen; Otherwise, if C < W, "Competition for adult females" is chosen. W is a parameter established before the optimization process.

Table 1. CEC 2017 functions

No	Function	FI
	Unimodal Functions	
1	Shifted and Rotated Bent Cigar Function	100
2	Shifted and Rotated Sum of Different Power Function	200
3	Shifted and Rotated Zakharov Function	300
	Simple Multimodal Functions	
4	Shifted and Rotated Rosenbrock's Function	400
5	Shifted and Rotated Rastrigin's Function	500
6	Shifted and Rotated Expanded Scaffer's Function	600
7	Shifted and Rotated Lunacek Bi_RastriginFunction	700
8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
9	Shifted and Rotated Levy Function	900
10	Shifted and Rotated Schwefel's Function	1000
	Hybrid functions	
11	Hybrid Function 1 (N = 3)	1100
12	Hybrid Function 2 (N = 3)	1200
13	Hybrid Function 3 (N = 3)	1300
14	Hybrid Function 4 (N = 4)	1400
15	Hybrid Function 5 (N = 4)	1500
16	Hybrid Function 6 (N = 4)	1600
17	Hybrid Function 6 (N = 5)	1700
18	Hybrid Function 6 (N = 5)	1800
19	Hybrid Function 6 (N = 5)	1900
20	Hybrid Function 6 (N = 6)	2000
	Composition Functions	
21	Composition Function 1 (N = 3)	2100
22	Composition Function 2 (N = 3)	2200
23	Composition Function 3 (N = 4)	2300
24	Composition Function 4 (N = 4)	2400
25	Composition Function 5 (N = 5)	2500
26	Composition Function 6 (N = 3)	2600
27	Composition Function 7 (N = 6)	2700
28	Composition Function 8 (N = 3)	2800
29	Composition Function 9 (N = 3)	2900
30	Composition Function 10 (N = 3)	3000

2.1.1 Following the Silverback

With the newly formed group, the silverback gorilla is young and in good health, like the other males in the group that faithfully follow him.

They all obey the silverback's orders to go to various areas in search of food sources. Additionally, members can influence the movement of the group. This approach is chosen when $C \geq W$. Eq. (7) is used to replicate this behavior:

$$GX(t+1) = L \times M \times (X(t) - rX_{siverback}) + X(t), \tag{7}$$

$$M = \left(\left| \sum_{i=1}^{N} GX(t) \right|^{g} \right)^{\frac{1}{g}}, \tag{8}$$

$$g = 2^{L}. \tag{9}$$

In (7), $X(t)$ represents the position vector of the gorilla, while X silverback is the position vector of the silverback gorilla. Furthermore, L is calculated using (4) and M using (8) where $GXi(t)$ shows the vector position of each candidate gorilla at iteration t. N denotes the total number of gorillas. g is also estimated using (9), and where L is also calculated using (4).

2.2 Stochastic Fractal Search (SFS) (Algorithm 2)

Fractals are formed by particles that move randomly, joining together to form a uniform pattern, each of the particles is added until they form a figure, the SFS has two important processes with which the method governs. The first process consists of particles diffusing near their position to comply with the intensification property, to find the global optimum and to avoid the local minimum.

This process is known as diffusion. The next consists of updating the position of a particle based on the position of the other particles in the group, the best particle produced is considered and the others will be discarded, this process also leads us to the diversification and Gaussian distribution for the random walk of particles and their growth in the diffusion process. The equations used in each of the aforementioned processes are explained below:

Table 2. GTO Triangular membership function

Triangular Membership Function					
Function	**F1**	**F2**	**F3**	**F4**	**F5**
Best	2.01×10^{4}	7.02×10^{15}	8.90×10^{3}	4.77×10^{2}	6.55×10^{2}
Worst	7.36×10^{6}	7.08×10^{26}	3.01×10^{4}	6.01×10^{2}	8.23×10^{2}
Average	9.56×10^{5}	3.91×10^{25}	1.88×10^{4}	5.27×10^{2}	7.74×10^{2}
STD	1.50×10^{6}	1.49×10^{26}	5.18×10^{3}	2.99×10^{1}	4.66×10^{1}
Function	**F6**	**F7**	**F8**	**F9**	**F10**
Best	6.34×10^{2}	9.54×10^{2}	9.03×10^{2}	3.63×10^{3}	4.35×10^{3}
Worst	6.66×10^{2}	1.35×10^{3}	1.03×10^{3}	5.87×10^{3}	9.28×10^{3}
Average	6.56×10^{2}	1.19×10^{3}	9.76×10^{2}	5.09×10^{3}	6.12×10^{3}
STD	8.59×10^{0}	1.02×10^{2}	3.05×10^{1}	5.16×10^{2}	1.37×10^{3}
Function	**F11**	**F12**	**F13**	**F14**	**F15**
Best	1.17×10^{3}	1.13×10^{5}	4.05×10^{3}	1.82×10^{3}	1.83×10^{3}
Worst	1.38×10^{3}	6.47×10^{6}	1.64×10^{5}	5.04×10^{4}	3.82×10^{4}
Average	1.25×10^{3}	2.24×10^{6}	2.49×10^{4}	1.16×10^{4}	1.16×10^{4}
STD	4.68×10^{1}	1.33×10^{6}	3.27×10^{4}	1.27×10^{4}	9.89×10^{3}
Function	**F16**	**F7**	**F18**	**F19**	**F20**
Best	1.92×10^{3}	1.89×10^{3}	3.15×10^{4}	2.03×10^{3}	2.31×10^{3}
Worst	3.52×10^{3}	2.93×10^{3}	3.15×10^{6}	1.88×10^{4}	3.20×10^{3}
Average	2.79×10^{3}	2.38×10^{3}	2.04×10^{5}	6.76×10^{3}	2.72×10^{3}
STD	3.76×10^{2}	2.89×10^{2}	2.57×10^{5}	5.16×10^{3}	2.15×10^{2}
Function	**F21**	**F22**	**F23**	**F24**	**F25**
Best	2.22×10^{3}	2.31×10^{3}	2.80×10^{3}	2.92×10^{3}	2.89×10^{3}
Worst	2.54×10^{3}	9.76×10^{3}	3.15×10^{3}	3.47×10^{3}	2.95×10^{3}
Average	2.44×10^{3}	4.07×10^{3}	2.91×10^{3}	3.07×10^{3}	2.92×10^{3}
STD	5.76×10^{1}	2.62×10^{3}	8.17×10^{1}	1.24×10^{2}	1.57×10^{1}
Function	**F26**	**F27**	**F28**	**F29**	**F30**
Best	3.21×10^{3}	3.24×10^{3}	3.21×10^{3}	2.81×10^{3}	9.09×10^{3}
Worst	8.31×10^{3}	3.71×10^{3}	3.33×10^{3}	5.96×10^{3}	7.05×10^{5}
Average	6.14×10^{3}	3.39×10^{3}	3.28×10^{3}	6.42×10^{3}	5.59×10^{4}
STD	1.45×10^{3}	1.14×10^{2}	2.82×10^{1}	4.46×10^{2}	1.24×10^{5}

$$P = LB + \varepsilon * (UP - LB). \tag{10}$$

The particle population P is randomly generated based on the problem constraints after setting the lower (LB) and the upper (UB) limits, where ε is a random number in the range [0,1].

The diffusion process (exploitation in fractal search) is expressed as follows:

$$GW_1 = \text{Gaussian}(\mu_{BP}, \sigma) + \left(\varepsilon * BP - \varepsilon',* P_i\right), \tag{11}$$

$$GW_2 = \text{Gaussian}\ (\mu_P, \sigma), \tag{12}$$

where ε, ε' represent random numbers in the range [0,1], $BP$is the best position of the point, i-th point in the group is represented by $P_i$and Gaussians a function that generates a random number from the normal distribution with a mean μ parameter and a standard deviation parameter σ:

$$\sigma = \frac{\log g}{g} \times |\, P_i - \ BP| \tag{13}$$

Table 3. GTO Gaussian membership function

Gaussian Membership Function					
Function	**F1**	**F2**	**F3**	**F4**	**F5**
Best	$3.75{\times}10^{4}$	$2.43{\times}10^{15}$	$8.72{\times}10^{3}$	$4.75{\times}10^{2}$	$6.18{\times}10^{2}$
Worst	$6.36{\times}10^{6}$	$3.22{\times}10^{28}$	$3.54{\times}10^{4}$	$6.35{\times}10^{2}$	$8.21{\times}10^{2}$
Average	$1.20{\times}10^{6}$	$1.14{\times}10^{27}$	$1.92{\times}10^{4}$	$5.22{\times}10^{2}$	$7.42{\times}10^{2}$
STD	$1.45{\times}10^{6}$	$5.87{\times}10^{27}$	$7.08{\times}10^{3}$	$3.09{\times}10^{1}$	$5.65{\times}10^{1}$
Function	**F6**	**F7**	**F8**	**F9**	**F10**
Best	$1.06{\times}10^{3}$	$9.19{\times}10^{2}$	$3.03{\times}10^{3}$	$4.73{\times}10^{3}$	$1.06{\times}10^{3}$
Worst	$1.37{\times}10^{3}$	$1.03{\times}10^{3}$	$5.78{\times}10^{3}$	$8.20{\times}10^{3}$	$1.37{\times}10^{3}$
Average	$1.22{\times}10^{3}$	$9.80{\times}10^{2}$	$5.12{\times}10^{3}$	$6.45{\times}10^{3}$	$1.22{\times}10^{3}$
STD	$8.85{\times}10^{1}$	$3.16{\times}10^{1}$	$5.51{\times}10^{2}$	$8.67{\times}10^{2}$	$8.85{\times}10^{1}$
Function	**F11**	**F12**	**F13**	**F14**	**F15**
Best	$1.18{\times}10^{3}$	$2.10{\times}10^{5}$	$3.74{\times}10^{3}$	$1.65{\times}10^{3}$	$2.02{\times}10^{3}$
Worst	$1.40{\times}10^{3}$	$6.99{\times}10^{6}$	$4.46{\times}10^{6}$	$3.62{\times}10^{4}$	$3.12{\times}10^{4}$
Average	$1.26{\times}10^{3}$	$1.91{\times}10^{6}$	$1.93{\times}10^{5}$	$9.51{\times}10^{3}$	$8.03{\times}10^{3}$
STD	$4.85{\times}10^{1}$	$1.61{\times}10^{6}$	$8.14{\times}10^{5}$	$9.76{\times}10^{3}$	$6.59{\times}10^{3}$
Function	**F16**	**F7**	**F18**	**F19**	**F20**
Best	$2.32{\times}10^{3}$	$1.82{\times}10^{3}$	$2.72{\times}10^{4}$	$2.08{\times}10^{3}$	$2.35{\times}10^{3}$
Worst	$3.80{\times}10^{3}$	$2.84{\times}10^{3}$	$1.02{\times}10^{6}$	$2.03{\times}10^{4}$	$3.25{\times}10^{3}$
Average	$2.98{\times}10^{3}$	$2.37{\times}10^{3}$	$1.68{\times}10^{5}$	$7.09{\times}10^{3}$	$2.63{\times}10^{3}$
STD	$3.76{\times}10^{2}$	$2.74{\times}10^{2}$	$1.87{\times}10^{5}$	$4.63{\times}10^{3}$	$2.13{\times}10^{2}$
Function	**F21**	**F22**	**F23**	**F24**	**F25**
Best	$2.41{\times}10^{3}$	$2.30{\times}10^{3}$	$2.79{\times}10^{3}$	$2.93{\times}10^{3}$	$2.89{\times}10^{3}$
Worst	$2.56{\times}10^{3}$	$1.02{\times}10^{4}$	$3.08{\times}10^{3}$	$4.06{\times}10^{3}$	$2.96{\times}10^{3}$
Average	$2.46{\times}10^{3}$	$4.97{\times}10^{3}$	$2.90{\times}10^{3}$	$3.10{\times}10^{3}$	$2.92{\times}10^{3}$
STD	$4.37{\times}10^{1}$	$2.99{\times}10^{3}$	$7.31{\times}10^{1}$	$2.11{\times}10^{2}$	$1.94{\times}10^{1}$
Function	**F26**	**F27**	**F28**	**F29**	**F30**
Best	$2.93{\times}10^{3}$	$3.21{\times}10^{3}$	$3.24{\times}10^{3}$	$3.78{\times}10^{3}$	$8.02{\times}10^{3}$
Worst	$8.17{\times}10^{3}$	$3.77{\times}10^{3}$	$3.37{\times}10^{3}$	$5.13{\times}10^{3}$	$2.43{\times}10^{7}$
Average	$5.76{\times}10^{3}$	$3.35{\times}10^{3}$	$3.29{\times}10^{3}$	$4.42{\times}10^{3}$	$8.60{\times}10^{5}$
STD	$1.45{\times}10^{3}$	$1.17{\times}10^{2}$	$2.77{\times}10^{1}$	$3.50{\times}10^{2}$	$4.43{\times}10^{6}$

where $\log g \ / \ g$ tends to a zero value as the generation number g increases. The update process (fractal search exploration) is:

$$\mathrm{Pa_i} = \frac{\mathrm{rank(P_i\,)}}{\mathrm{N}}, \tag{14}$$

where N is the number of particles in the group and Pa_i is the estimated probability value for a particle whose rank among the others is given by the "rank" function.

The particles are classified according to their fitness value and then a probability value is given to each particle *i*:

$$P_i'(j) = P_x(j) - \varepsilon * \left(P_y\,(j) - P_i(j)\right), \tag{15}$$

where the augmented component is represented by $P_i'(j)$, and P_x, P_y are different points randomly chosen from the group. P_i' replacements P_i if it has a better fitness value:

$$P_i' = \ P_i - \ \varepsilon * (P_x \ - BP) |\ \acute{\varepsilon} \leq 0.5 \ , \tag{16}$$

Table 4. SFS Gaussian membership function

Gaussian Membership Function					
Function	**F1**	**F2**	**F3**	**F4**	**F5**
Best	4.97×10^{4}	2.63×10^{36}	5.77×10^{4}	4.87×10^{2}	7.60×10^{2}
Worst	3.70×10^{5}	3.87×10^{40}	6.17×10^{4}	6.56×10^{2}	8.46×10^{2}
Average	1.58×10^{5}	2.07×10^{39}	5.54×10^{4}	5.56×10^{2}	7.84×10^{2}
STD	7.35×10^{4}	7.57×10^{39}	8.70×10^{3}	4.23×10^{1}	2.91×10^{1}
Function	**F6**	**F7**	**F8**	**F9**	**F10**
Best	6.00×10^{2}	9.96×10^{2}	1.02×10^{3}	9.47×10^{2}	1.07×10^{4}
Worst	6.01×10^{2}	1.03×10^{3}	1.11×10^{3}	1.15×10^{3}	1.18×10^{4}
Average	6.01×10^{2}	1.06×10^{3}	1.08×10^{3}	1.10×10^{3}	1.12×10^{4}
STD	2.05×10^{-1}	2.58×10^{1}	3.09×10^{1}	1.24×10^{2}	5.78×10^{2}
Function	**F11**	**F12**	**F13**	**F14**	**F15**
Best	1.32×10^{3}	2.12×10^{6}	2.26×10^{3}	1.93×10^{3}	3.88×10^{3}
Worst	1.42×10^{3}	9.29×10^{6}	1.00×10^{5}	2.00×10^{3}	9.75×10^{3}
Average	1.36×10^{3}	3.77×10^{6}	2.19×10^{4}	1.91×10^{3}	4.64×10^{3}
STD	2.97×10^{1}	1.50×10^{6}	1.93×10^{4}	1.04×10^{2}	1.39×10^{3}
Function	**F16**	**F7**	**F18**	**F19**	**F20**
Best	3.07×10^{3}	2.91×10^{3}	2.57×10^{4}	3.46×10^{3}	2.77×10^{3}
Worst	4.01×10^{3}	3.42×10^{3}	1.26×10^{5}	1.56×10^{4}	3.03×10^{3}
Average	3.65×10^{3}	3.15×10^{3}	5.32×10^{4}	6.00×10^{3}	3.15×10^{3}
STD	3.02×10^{2}	2.20×10^{2}	2.20×10^{4}	2.49×10^{3}	2.23×10^{2}
Function	**F21**	**F22**	**F23**	**F24**	**F25**
Best	2.54×10^{3}	2.31×10^{3}	1.32×10^{4}	3.09×10^{3}	3.03×10^{3}
Worst	2.58×10^{3}	1.32×10^{4}	3.05×10^{3}	3.22×10^{3}	3.11×10^{3}
Average	2.57×10^{3}	1.12×10^{4}	3.00×10^{3}	3.17×10^{3}	3.06×10^{3}
STD	2.95×10^{1}	4.16×10^{3}	3.22×10^{1}	3.78×10^{1}	2.19×10^{1}
Function	**F26**	**F27**	**F28**	**F29**	**F30**
Best	5.82×10^{3}	3.36×10^{3}	3.32×10^{3}	3.60×10^{3}	2.93×10^{6}
Worst	7.01×10^{3}	3.38×10^{3}	3.29×10^{3}	4.42×10^{3}	5.08×10^{6}
Average	6.39×10^{3}	3.43×10^{3}	3.35×10^{3}	4.18×10^{3}	3.95×10^{6}
STD	3.65×10^{2}	3.85×10^{1}	2.92×10^{1}	2.42×10^{2}	7.45×10^{5}

$$P_i' = P_i - \varepsilon * (P_x - P_y)| \text{ otherwise.} \tag{17}$$

At the end of the first update process, the second one begins by ranking all the resulting points once more based on Eqs. (16) and (17).

As before, if Pa_i is less than a random number ε, the current point, P_i is modified by using the previous equations, where the $x$and y indices must be different. Of course, the new pointP_i' is replaced by P_i if it is better than P_i.

2.3 Fuzzy Logic

Fuzzy logic developed by Lotfi Zadeh [31] is a type of mathematical logic that allows for handling ambiguous or imprecise concepts. Unlike classical logic, which assumes that an object or a proposition is true or false in an exclusive manner, fuzzy logic allows us to represent the degree of membership or veracity of an object in a fuzzy set. In fuzzy logic, truth values can be any number

Table 5. Parameters used in Z-test for GTO VS SFS

Parameter of Z-test for GTO vs SFS	
Critical Value (Zc)	1.64
Confidence interval	95%
H_0	$\mu 1 \geq \mu 2$
H_a (Claim)	$\mu 1 < \mu 2$
Alpha	0.05

between 0 and 1, meaning that an object can have a degree of partial membership in a set rather than simply being true or false.

This allows modelling and representing situations in which uncertainty, imprecision or subjectivity are present. Fuzzy logic has been successfully applied in various fields such as artificial intelligence, system control, decision-making, pattern recognition and engineering, among others. Its applications are based on the ability to handle and reason with incomplete or uncertain information as described by the following authors.[32, 33, 34, 35].

A fuzzy inference system is a system that uses fuzzy logic to perform reasoning and decision-making based on uncertainty and imprecision of data. A fuzzy inference system consists of three main components:

1 The fuzzy knowledge base: it is where the rules that relate the input variables to the output variables are defined.

2 These rules are formulated in linguistic terms and are based on expert knowledge of the domain.

3 The fuzzy inference engine: it is responsible for combining the rules of the knowledge base and calculating the output in fuzzy terms. It uses different inference methods, such as the Mamdani method or the maximum method.

4 The fuzzy database: contains the input information necessary for the system to make inferences. The input variables are represented by fuzzy sets, which assign a degree of membership to each possible value.

The fuzzy inference system can be used in various fields, such as engineering, medicine, robotics, and control systems, among others. Its ability to handle imprecision and uncertainty makes it especially useful when available data is incomplete or ambiguous.[36, 37].

3 Comparison Analysis with Dynamic Parameter Adjustment in the Bio-Inspired Methods

The CEC 2017 mathematical functions are sets of known problems, which are used to evaluate optimization algorithms. These functions provide a common framework for comparing the performance of different algorithms [38, 39].

Table 1 shows the mathematical functions of the CEC 2017, which are classified as unimodal, multimodal, hybrid and composite functions. The number of functions is 30, each of which has a different global value.

These values are what the algorithms that undergo this type of testing must find since they are the way to evaluate the behavior and effectiveness of the methods. Next, the dynamic parameter adaptation process used to optimize the membership functions of the fuzzy systems used is described.

In the first instance, it begins with the analysis of the problem, in this case, it corresponds to dynamically adapting the values of the parameters of the membership functions to improve the performance of the method in the search for the global optimum for each mathematical function.

Afterwards, the construction of the fuzzy system continues, for which the optimization method is explored to know the variables to which said adjustment is applied. Once found, they are used as outputs of the fuzzy system, and the iteration is used at the input, so that, in each iteration (each time this occurs in the method), the dynamic adaptation to the chosen variables would be performed.

Figure 4 shows the fuzzy system used for the dynamic parameters adaptation in the GTO method. For this first case study, triangular membership functions are used to analyze its operation. Figure 5 shows the fuzzy system used for dynamic parameter adaptation also implemented in the GTO method.

In this second case study, Gaussian membership functions are used. It should be noted

ISSN 2007-9737

Table 6. Z-test results

	Calculated z	Evidence
1	3,931	not significan
2	-1,498	not significan
3	-17,677	significan
4	-3,555	significan
5	-3.62	significan
6	46,611	not significan
7	9,507	not significan
8	-12,393	significan
9	38,986	not significan
10	-24,968	significan
11	-9,631	significan
12	-4.63	significan
13	1,151	not significan
14	4,265	not significan
15	2,757	not significan
16	-7,609	significan
17	-12,158	significan
18	3,339	not significan
19	1,136	not significan
20	-9,236	significan
21	-11,427	significan
22	-6,661	significan
23	-6,857	significan
24	-1,789	significan
25	-26.21	significan
26	-2,308	significan
27	-3,557	significan
28	-8,165	significan
29	3,089	not significan
30	-3,768	significan

that both fuzzy systems use the iterations as input and adjust the parameters of p and $\boldsymbol{W}$, as outputs.

Figure 6 shows the fuzzy system used for the dynamic parameters adaptation also implemented in the SFS method, using the iteration as input and the diffusion parameter as output. For this case study, triangular membership functions are used. The fuzzy rules are listed as follows:

1 If iteration is Low then p is High and w is MediumLow.

2 If iteration is MediumLow then p is Medium and w is Medium

3 If iteration is Medium then p is MediumLow and w is MediumHigh.

4 If iteration is MediumHigh then p is Low and w is High.

5 If iteration is High then p is Low and w is High.

To execute the methods, the following architecture was used: In the GTO alpha 0.1 algorithm, delta 0.1, agents 50, Iterations 500, dimensions 50 and for SFS agents 50, Iterations 500, dimensions 50.

4 Results

Table 2 presents the results derived from the use of a fuzzy system with triangular membership functions. This system was developed to improve the search for the global optimum in the CEC 2017 functions.

The first column describes the metrics used to analyze the results. It starts with the function evaluated, followed by the best result obtained for that function about the number of experiments established.

Subsequently, the worst result obtained under the same number of experiments is presented. Likewise, the average is offered as a measure of central tendency that indicates the center of the results within the statistical distribution.

Finally, the standard deviation is shown, which indicates the dispersion of the data in relation to the mean. In particular, in the functions *f4, f5, f6, f7* and *f8*, the values obtained were very close to the corresponding optima of those function.

Table 3 presents the results obtained by using the fuzzy system, this time developed with Gaussian membership functions. This variation was carried out to determine which of the systems offers better performance when searching for the values of mathematical functions.

The structure and content of the table are similar to those of Table 2, which details the metrics used to analyze the results, including the function evaluated, the best and worst result, the average and the standard deviation, giving the best results in the functions f4, f5, f6 and f8 approaching the global optimum.

This approach of employing Gaussian membership functions represents an additional exploration to determine the effectiveness of the fuzzy system in optimizing mathematical functions. The results of the SFS method with dynamic parameter adaptation are detailed in Table 4.

This table shows the values corresponding to the best result obtained, the worst result, the average, and the standard deviation, presented in a format similar to the previous tables, to provide a complete view of the performance of the method.

Statistical test.

The Z parametric test is used to perform the statistical analysis, to compare the results obtained throughout the experimentation. Mathematically, the statistical test is expressed as:

$$Z = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}, \tag{18}$$

were, $\bar{x}_1 - \bar{x}_2$ it represents the difference between the sample means, and $\mu_1 - \mu_2$ denotes the difference between the population means:

$$\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}. \tag{19}$$

Represents the sum of the population standard deviations, and (n_1, n_2) represents the sample size. In the experiments carried out with the mathematical functions of CEC2017, where Type-1 fuzzy systems are used for the adaptation of dynamic parameters, the following hypotheses are established:

– Null hypothesis (Ho): The results provided by the GTO using dynamic parameter tuning with Gaussian membership functions are greater than or equal to those of the SFS method using dynamic parameter adaptation with Gaussian membership functions.

– Alternative hypothesis (Ha): The results provided by the GTO using dynamic parameter adjustment with Gaussian membership functions are lower than those obtained by the SFS method using dynamic parameter adaptation with Trapezoidal membership functions. Table 5 presents the statistical parameters used for this analysis:

The results of the Z test applied to the thirty CEC2017 functions are shown in Table 3. Columns 2 and 3 show the results of the GTO using Gaussian membership functions and their standard deviation, while columns 4 and 5 present the results of the SFS using Gaussian membership functions.

Column 6 describes the results of the Z test, and column 7 indicates whether there is (Y) or not (NS) significant evidence to reject the null hypothesis. It can be seen that in 19 of the 30 functions, there is evidence that supports the claim that the GTO method dynamic adjustment with Gaussian membership functions provides better results, therefore it can be said that the GTO method is capable of adapting to obtain the optimum of each function demonstrating its effectiveness when adjusted with Type-1 fuzzy logic.

In Table 6 you can see column 1 where the benchmark functions are shown, column 2 corresponds to the Z calculated which refers to the value obtained when performing a hypothesis test, this is used to evaluate statements about the mean of a population when the standard deviation is known.

The "critical z" is the critical value of the z statistic that is used in hypothesis testing to establish a boundary between the rejection region and the non-rejection region, to all functions $z = -1.64$. In hypothesis testing, the "calculated z" is usually compared to the "critical z" to make decisions about whether or not to reject the null hypothesis.

5 Discussion

The purpose of this study is to compare the adaptive capacity of the GTO and SFS methods to identify global optima in the CEC 2017 functions. Type-1 fuzzy logic was used for dynamically adapting the parameters of the membership functions, seeking to improve the performance of both methods.

The mathematical functions evaluated have varying degrees of complexity, resulting in different global optima. This diversity challenges the methods and demonstrates their performance and effectiveness in solving complex problems.

After meticulously analyzing each method, variables that required dynamic adjustments were identified. Fuzzy logic, widely used in solving optimization problems with bio-inspired algorithms, was crucial to determine which method was best suited and offered the best results.

When observing the results in Tables 2 and 3, it is noted that the results obtained by the GTO method, using triangular membership functions, and the fuzzy system using Gaussian functions, showed significant similarities.

This suggests that the difference between both systems was minimal, demonstrating a similar adaptability of the dynamically adjusted method in each iteration, which supports its effectiveness in performance with dynamic parameter adjustment.

Performing dynamic parameter adaptation with Type-1 fuzzy logic has been essential to improve the approximation of the methods to the optimal values of the functions, despite its complexity. However, the SFS algorithm, as detailed in Table 4, faced more difficulties in reaching the optimal values.

Their results were found to be further from the real values, indicating that their adaptability with dynamic parameter adaptation was not as effective as in the case of the GTO method.

6 Conclusions and Future Work

In this article, a comprehensive case study was conducted that compared two bioinspired methods to analyze their performance and adaptability. Benchmark mathematical functions from CEC 2017 were used as evaluation tools, which present different levels of complexity and, therefore, different global optima.

The results obtained were highly positive. The GTO method, which uses a fuzzy system with Gaussian membership functions and triangular membership functions, showed significant improvements in functions 4, 5, 6, 7 and 8, as detailed in tables 3 and 4. On the other hand, the FSF method demonstrated good results in functions 4, 5, 6, 7 and 9, as seen in Table 5.

The results obtained reflect the effectiveness of both methods to find the optimum in some of the evaluated functions. This comparative analysis provides a clear view of the relative performance of the algorithms in different contexts and lays the foundation for future research and improvements in the optimization of specific functions.

The evaluation of the FSF and GTO methods was carried out using Type-1 fuzzy logic, to adjust the parameters of the membership functions used in the fuzzy controller, this optimization was carried out to know the behavior of the methods when searching for the global of each function, where satisfactory results were obtained, although they can be improved using other intelligent computing techniques such as type 2 fuzzy logic, like in [40-42].Type-2 fuzzy logic will help us have a better insertion threshold and find better results.

It is intended to use different membership functions, in addition to adding more inputs to fuzzy systems to improve their performance, such as diversity.

Some type of hybridization between the methods can also be carried out since both have proven to be efficient for specific optimizations. In order to carry out some type of hybridization, it will be necessary to thoroughly analyze each part of the algorithms, especially those parameters that help their convergence and performance.

References

1. **Holland, J. H. (1967).** Genetic algorithms understand genetic algorithms. Surprise, Vol. 96, No. 1, pp. 12–15.
2. **Pal, S. K., Wang, P. P. (1996).** Genetic algorithms for pattern recognition. CRC Press.
3. **Wu, D., Tan, W. W. (2006).** Genetic learning and performance evaluation of interval Type-2 fuzzy logic controllers. Engineering Applications of Artificial Intelligence, Vol. 19, No. 8, pp. 829–841. DOI: 10.1016/j.engappai.2005.12.011.
4. **Babaie-Kafaki, S., Ghanbari, R., Mahdavi-Amiri, N. (2016).** Hybridizations of genetic algorithms and neighborhood search metaheuristics for fuzzy bus terminal location problems. Applied Soft Computing, Vol. 46, pp. 220—229. DOI: 10.1016/j.asoc.2016.03.005.
5. **Luthra, J., Pal, S. K. (2011).** A hybrid firefly algorithm using genetic operators for the cryptanalysis of a monoalphabetic substitution

cipher. World Congress on Information and Communication Technologies, pp. 202—206. DOI: 10.1109/wict.2011.6141244.

6. **Han, S., Pedrycz, W., Han, C. (2005).** Nonlinear channel blind equalization using hybrid genetic algorithm with simulated annealing. Mathematical and Computer Modelling, Vol. 41, No. 6-7, pp. 697–709. DOI: 10.1016/j.mcm.2004.05.006.
7. **Wagner, C., Hagras, H. (2007)**. A genetic algorithm based architecture for evolving Type-2 fuzzy logic controllers for real world autonomous mobile robots. IEEE International Fuzzy Systems Conference, pp. 1–6. DOI: 10.1109/fuzzy.2007.4295364.
8. **Trrad, I., Smadi, T. A., Al_Wahshat, H. (2019).** Application of fuzzy logic to cognitive wireless communications. International Journal of Recent Technology and Engineering (IJRTE), Vol. 8, No. 3, pp. 2228–2234. DOI: 10.35940/ijrte.b2065.098319.
9. **Pratiwi, W., Sofwan, A., Setiawan, I. (2020).** Implementation of fuzzy logic method for automation of decision making of Boeing aircraft landing. IAES International Journal of Artificial Intelligence, Vol. 10, No. 3. DOI: 10.11 591/ijai.v10.i3.
10. **Yang, X. S., He, X. (2013).** Firefly algorithm: recent advances and applications. International Journal of Swarm Intelligence, Vol. 1, No. 1, pp. 36–50. DOI: 10.1504/ijsi. 2013.055801.
11. **Yang, X. (2009).** Firefly algorithm, Lévy flights and global optimization. Research and Development in Intelligent Systems XXVI, pp. 209–218. DOI: 10.1007/978-1-84882-983- 1_15.
12. **Lagunes, M. L., Castillo, O., Soria, J., Garcia, M., Valdez, F. (2018).** Optimization of granulation for fuzzy controllers of autonomous mobile robots using the firefly algorithm. Granular Computing, Vol. 4, No. 2, pp. 185–195. DOI: 10.1007/s41066-018-0121- 6.
13. **Lagunes, M. L., Castillo, O., Soria, J., Garcia, M., Valdez, F. (2018).** Optimization of granulation for fuzzy controllers of autonomous mobile robots using the firefly algorithm. Granular Computing, Vol. 4, No. 2, pp. 185–195. DOI: 10.1007/s41066-018-0121- 6.
14. **Lagunes, M. L., Castillo, O., Valdez, F., Soria, J., Melin, P. (2018)**. Parameter optimization for membership functions of type-2 fuzzy controllers for autonomous mobile robots using the firefly algorithm. Fuzzy Information Processing, pp. 569–579. DOI: 10.1007/978-3-319-95312-0_50.
15. **Miramontes, I., Guzman, J., Melin, P., Prado-Arechiga, G. (2018).** Optimal design of interval type-2 fuzzy heart rate level classification systems using the bird swarm algorithm. Algorithms, Vol. 11, No. 12, pp. 206. DOI: 10.3390/a11120206.
16. **Tarik, R., Adil, B. (2022).** Ant colony optimization algorithm and fuzzy logic for switched reluctance generator control. AIMS Energy, Vol. 10, No. 5, pp. 987–1004. DOI: 10.3934/energy.2022045.
17. **Mohamed, A., Saber, W., Elnahry, I., Hassanien, A. E. (2020).** Coyote optimization based on a fuzzy logic algorithm for energy-efficiency in wireless sensor networks. IEEE Access, Vol. 8, pp. 185816–185829. DOI: 10. 1109/access.2020.3029683.
18. **Bhoi, A. K., Mallick, P. K., Liu, C. M., Balas, V. E. (Eds.). (2021).** Bio-inspired neurocomputing. Berlin/Heidelberg, Germany: Springer, Vol. 310. DOI: 10.1007/978-981-15-5495-7.
19. **Miramontes, I., Melin, P. (2022).** Interval type-2 fuzzy approach for dynamic parameter adaptation in the bird swarm algorithm for the optimization of fuzzy medical classifier. Axioms, Vol. 11, No. 9, pp. 485. DOI: 10.3390/ axioms11090485.
20. **Miramontes, I., Melin, P. (2023).** Enhancing dynamic parameter adaptation in the bird swarm algorithm using general type-2 fuzzy analysis and mathematical functions. Axioms, Vol. 12, No. 9, pp. 834. DOI: 10.3390/ axioms12090834.
21. **Glover, F. (1990).** Tabu search: A tutorial. Interfaces, Vol. 20, No. 4, pp. 74–94. DOI: 10. 1287/inte.20.4.74.

22. **Dorigo, M., Stützle, T. (2003).** The ant colony optimization metaheuristic: algorithms, applications, and advances. Handbook of Metaheuristics, pp. 250–285. DOI: 10.1007/0-306-48056-5_9.
23. **Xu, Y., Fan, P., Yuan, L. (2013).** A simple and efficient artificial bee colony algorithm. Mathematical Problems in Engineering, Vol. 2013, pp. 1–9. DOI: 10.1155/2013/526315.
24. **Patwal, R. S., Narang, N., Garg, H. (2018).** A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. Energy, Vol. 142, pp. 822–837. DOI: 10.1016/j.energy.2017.10.052.
25. **Brito, J., Martínez, F., Moreno, J., Verdegay, J. (2015).** An ACO hybrid metaheuristic for close–open vehicle routing problems with time windows and fuzzy constraints. Applied Soft Computing, Vol. 32, pp. 154–163. DOI: 10.10 16/j.asoc.2015.03.026.
26. **Gomaa, I. H., Zaher, H., Ragaa-Saeid, N., Sayed, H. (2023).** A novel enhanced gorilla troops optimizer algorithm for global optimization problems. International Journal of Industrial Engineering & Production Research, Vol. 34, No. 1, pp. 1–9. DOI: 10.22068/ ijiepr.34.1.3.
27. **Salimi, H. (2015).** Stochastic fractal search: A powerful metaheuristic algorithm. Knowledge-Based Systems, Vol. 75, pp. 1–18. DOI: 10.10 16/j.knosys.2014.07.025.
28. **Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P., Definitions, P. (2016).** Evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. Technical Report, pp. 5–8.
29. **Sayed, G. I., Hassanien, A. E. (2021).** A novel chaotic artificial gorilla troops optimizer and its application for fundus images segmentation. Lecture Notes on Data Engineering and Communications Technologies, pp. 318–329. DOI: 10.1007/978-3-030-89701-7_28.
30. **Abdollahzadeh, B., Gharehchopogh, F. S., Mirjalili, S. (2021).** Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. International Journal of Intelligent Systems, Vol. 36, No. 10, pp. 5887--5958. DOI: 10.10 02/int.22535.
31. **Zadeh, L. (1988).** Fuzzy logic. Computer, Vol. 21, No. 4, pp. 83–93. DOI: 10.1109/2.53.
32. **Karnik, N., Mendel, J., Liang, Q. (1999).** Type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems, Vol. 7, No. 6, pp. 643–658. DOI: 10.1109/91.811231.
33. **Schouten, N., Salman, M., Kheir, N. (2002).** Fuzzy logic control for parallel hybrid vehicles. IEEE Transactions on Control Systems Technology, Vol. 10, No. 3, pp. 460–468. DOI: 10.1109/87.998036.
34. **Liang, Q., Mendel, J. (2000).** Interval type-2 fuzzy logic systems: theory and design. IEEE Transactions on Fuzzy Systems, Vol. 8, No. 5, pp. 535–550. DOI: 10.1109/91.873577.
35. **Mendel, J. M., John, R. I., Liu, F. (2006).** Interval type-2 fuzzy logic systems made simple. IEEE Transactions on Fuzzy Systems, Vol. 14, No. 6, pp. 808–821. DOI: 10.1109/ tfuzz.2006.879986.
36. **Zadeh, L. A. (2008).** Is there a need for fuzzy logic? Information Sciences, Vol. 178, No. 13, pp. 2751–2779. DOI: 10.1016/j.ins.2008. 02.012.
37. **Serrano-Guerrero, J., Romero, F. P., Olivas, J. A. (2021).** Fuzzy logic applied to opinion mining: a review. Knowledge-Based Systems, Vol. 222, pp. 107018. DOI: 10.1016/j.knosys. 2021.107018.
38. **Daoud, M. S., Shehab, M., Abualigah, L., Thanh, C. (2023).** Hybrid modified chimp optimization algorithm and reinforcement learning for global numeric optimization. Journal of Bionic Engineering, Vol. 20, No. 6, pp. 2896–2915. DOI: 10.1007/s42235-023-00394-2.
39. **Shehab, M., Abualigah, L. (2022).** Opposition-based learning multi-verse optimizer with disruption operator for optimization problems. Soft Computing, Vol. 26, No. 21, pp. 11669–11693. DOI: 10.1007/s00500-022-07470-5.
40. **Tai, K., El-Sayed, A., Biglarbegian, M., Gonzalez, C., Castillo, O., Mahmud, S. (2016).** Review of recent type-2 fuzzy

controller applications. Algorithms, Vol. 9, No. 2, pp. 39. DOI: 10.3390/a9020039.

41. **Ontiveros, E., Melin, P., Castillo, O. (2020).** Comparative study of interval type-2 and general type-2 fuzzy systems in medical diagnosis. Information Sciences, Vol. 525, pp. 37–53. DOI: 10.1016/j.ins.2020.03.059.
42. **Moreno, J. E., Sanchez, M. A., Mendoza, O., Rodríguez-Díaz, A., Castillo, O., Melin, P., Castro, J. R. (2020).** Design of an interval type-2 fuzzy model with justifiable uncertainty. Information Sciences, Vol. 513, pp. 206–221. DOI: 10.1016/j.ins.2019.10.042.
43. **Guerrero, M., Valdez, F., Castillo, O. (2022).** Comparative study between type-1 and interval type-2 fuzzy systems in parameter adaptation for the cuckoo search algorithm. Symmetry, Vol. 14, No. 11, pp. 2289. DOI: 10. 3390/sym14112289.
44. **Cuevas, F., Castillo, O., Cortés-Antonio, P. (2022).** Generalized type-2 fuzzy parameter adaptation in the marine predator algorithm for fuzzy controller parameterization in mobile robots. Symmetry, Vol. 14, No. 5, pp. 859. DOI: 10.3390/sym14050859.

Article received on 10/02/2024; accepted on 01/05/2024.
[*]Corresponding author is Marylu L. Lagunes.