

Type of the Paper (Article, Review, Communication, etc.)

# A new Approach for Dynamic Stochastic Fractal Search with Fuzzy Logic for Parameter Adaptation

Marylu L. Lagunes<sup>1</sup>, Oscar Castillo<sup>1\*</sup>, Fevrier Valdez<sup>1</sup>, Jose Soria<sup>1</sup>, Patricia Melin

<sup>1</sup> Tijuana Institute of Technology, Calzada Tecnológico s/n, Fracc. Tomas Aquino, 22379, Tijuana, Mexico

\* Correspondence: ocastillo@tectijuana.mx

**Abstract:** Metaheuristic algorithms are widely used as optimization methods, due to their global exploration and exploitation characteristics, which obtain better results than a simple heuristic. The Stochastic Fractal Search (SFS) is a novel method inspired by the process of stochastic growth in nature and the use of the fractal mathematical concept. Considering the chaotic-stochastic diffusion property, an improved Dynamic Stochastic Fractal Search (DSFS) optimization algorithm is presented. The DSFS algorithm was tested with benchmark functions, such as the multimodal, hybrid and composite functions, to evaluate the performance of the algorithm with dynamic parameter adaptation with type-1 and type-2 fuzzy inference models. The main contribution of the article is the utilization of fuzzy logic in the adaptation of the diffusion parameter in a dynamic fashion. This parameter is in charge of creating new fractal particles, and the diversity and iteration are the input information used in the fuzzy system to control the values of diffusion.

**Keywords:** Fractal Search; Fuzzy Logic; Parameter Adaptation, CEC'2017.

## 1. Introduction

Metaheuristic algorithms are applied to optimization problems due to their characteristics that help in searching for the global optimum, while simple heuristics are mostly capable in searching for local optimum and these are not very effective for optimization solutions. Stochastic methods have the peculiarity of being characterized by stochastic random variables. In the current literature several of the most popular stochastic algorithms for optimization can be found, like the Genetic Algorithm (GA) [1], [2] that is inspired by biological evolution with random genetic combinations and mutations in a chromosome, and is based on the selection, crossover, mutation and replacement operators. Particle Swarm Optimization (PSO) inspiration was the natural fish and birds behavior when moving in swarms, where each particle moves randomly to find the global optimum, updating its speed and position until finding the best global solution [3]. Tabu Search (TS) [4], [5] is an iterative method that builds meta-strategies to build a neighborhood and avoids getting trapped in local optima. As mentioned above, these algorithms are the most widely used, for example in [6] an algorithm for optimization of hybrid particle swarms incorporating chaos is proposed, where the adaptive inertia weight factor (AIWF) is used for enhancing PSO in balancing in an efficient way the diversification and intensification abilities of the algorithm. In this case, the hybridization of PSO with AIWF and chaos is used to build a chaotic PSO (CPSO), combining prudently the evolutionary search ability based on the population of PSO and the behavior of chaotic search. Furthermore [7]

proposes new PSO algorithm that rely on chaotic equation maps for parameter adaptation, this is done through the use of chaotic number generators every time the classical PSO algorithm needs a random number [8]. On the other hand, [9] develops an improved particle swarm optimization (IPSO) algorithm for enhancing the performance of the traditional PSO, which uses a dynamic inertia weight. In [10] the authors present an invasive weed optimization (IWO) metaheuristic, which is a weed colony based population optimization approach based also on chaos theory. In addition to improvements to stochastic optimization methods, there are also combinations or hybridizations with fuzzy logic to improve performance or reach a specific solution, for example the author in [11] an adaptive fuzzy control approach to control chaotic unified systems was proposed. Also in [12]–[15] a multi-metaheuristic model is developed for the optimal design of fuzzy controllers. An enhancement to the Bat algorithm is carried out in [16] for the dynamic adaptation of parameters. There are different applications using metaheuristics and fuzzy logic for optimization [17]–[19]. In this article we focus on testing the efficiency of the Dynamic Stochastic Fractal Search (DSFS) method in optimizing unimodal, multimodal, hybrid and composite functions. First, the Stochastic Fractal Search (SFS) algorithm is taken by analyzing its chaotic-stochastic characteristics in the diffusion process, where each of its particles is generated and moved in a random stochastic way, second, it is detected that the stochastic movement of each particle may not be optimal due to the formation of the fractal not being able to get to explore and exploit the entire search space. Therefore, diversity is introduced for each iteration, to eventually get closer to the global optima by looking for the particles with the best fitness, adapting an inference system for their adjustment. To obtain a comparison of the efficiency of the improved method, 30 functions with different dimensions were evaluated generating satisfactory results compared to other algorithms. The main contribution in this article was the improvement of the SFS method, since the algorithm had a disadvantage in the diffusion parameter because it only uses the Gaussian distribution as its randomness method, to compensate for the fact that the particles might not be able to cover all the space. search, it was decided to add, dynamic adjustment to said parameter, to achieve a better movement in each of the newly generated fractal particles, this improvement was implemented using type-1 and type-2 fuzzy systems, by making a dynamic changes with a controller, that has diversity as input 1 and iteration as input 2, diversification is charge of spreading the particles throughout the search area for each iteration, as a result, a dynamically adapted method is obtained that does not stagnate as fast in the optimal ones local, thus reaching the global optimum and therefore improving the effectiveness of the Dynamic Stochastic Fractal Search (DSFS) method.

The rest of this article is structured as indicated below. Section 2 puts forward the Stochastic Fractal Search (SFS) method. Section 3 outlines the proposed Dynamic Stochastic Fractal Search (DSFS). In Section 4 a summary of the experimental results is presented. In Section 5 the advantages of the method are highlighted with a discussion of the achieved results. Lastly, in Section 6 conclusions about the modified Dynamic Stochastic Fractal Search (DSFS) method are presented.

## 2. Materials and Methods Stochastic Fractal Search (SFS)

The term fractal was used for the first time by [20] Benoit Mandelbrot who described in his theory of fractals, geometric patterns generated in nature. There are some methods to generate fractals such as systems of iterated functions [21], Strange attractors [22], L-systems [23], finite subdivision rules [24] and random fractals [25]. On the part of the generation of random fractals we find the Diffusion Limited Aggregation (DLA), which consists of the formation of fractals starting with an initial particle that is called a seed and is usually situated at the origin [26], [27]. Then other particles are randomly generated near the origin causing diffusion. This diffusion process is carried out with a mathematical algorithm as a random walk where the diffuser particle adheres to the initial particle, this process is iteratively repeated and stops only when a group of particles is formed. While the group is forming, the probability of a particle getting stuck at the end has been incremented with respect to those that reach the interior, forming a cluster with a structure similar to a branch. These branches can shape chaotic-stochastic patterns, such as the formation of lightning in nature [28].

In Stochastic Fractal Search (SFS) [29] two important processes occur: the diffusion and the update, respectively. In the first one, the particles diffuse near their position to fulfill the intensification property (exploitation), with this the possibility of finding the global minima is increased, and at the same time avoiding getting stuck at a local minima. In the second one, a simulation of how one point is updating its position using the positions of other points in the group is made. In this process, the best particle produced from diffusion is the only one that is taken into account, and the remaining particles are eliminated. The equations used in each of the aforementioned processes are explained below.

$$P = LB + \varepsilon * (UP - LB) \quad (1)$$

The particle population  $P$  is randomly produced considering the problem constraints after setting the lower (LB) and the upper (UB) limits, where  $\varepsilon$  is a number randomly produced in the range [0,1].

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

$$GW_1 = \text{Gaussian}(\mu_{BP}, \sigma) + (\varepsilon * BP - \varepsilon' * P_i) \quad (2)$$

$$GW_2 = \text{Gaussian}(\mu_p, \sigma) \quad (3)$$

where  $\varepsilon$ ,  $\varepsilon'$  are numbers randomly generated in the range [0,1],  $BP$  represents the best position of the point,  $i$ -th indicates a point  $P_i$  and *Gaussian* represents a normal distribution that randomly generates numbers with a mean  $\mu$  and a standard deviation  $\sigma$ :

$$\sigma = \frac{\log g}{g} * |P_i - BP| \quad (4)$$

where  $\frac{\log g}{g}$  tends to a zero value as  $g$  increases.

The update process (representing exploration in fractal search) is:

$$Pa_i = \frac{\text{rank}(P_i)}{N} \quad (5)$$

where  $N$  represents the number of particles and  $Pa_i$  is the estimated particle probability, whose rank is given by the "rank" function. A classification of the particles is done according to their fitness value. Finally, a probability is assigned to each particle  $i$ .

$$P'_i(j) = P_x(j) - \varepsilon * (P_y(j) - P_i(j)) \quad (6)$$

where the augmented component is given by  $P'_i(j)$ , and  $P_x, P_y$  are different points selected from the group in a random fashion.  $P'_i$  replaces  $P_i$  if it achieves a better fitness value.

$$P'_i = P_i - \varepsilon * (P_x - BP) | \varepsilon' \leq 0.5 \quad (7)$$

$$P'_i = P_i - \varepsilon * (P_x - P_y) | \text{otherwise} \quad (8)$$

Once the first updating stage is finished, the second one initiates with a ranking of all points based on Eqs. (7) and (8). As previously mentioned, if  $Pa_i$  is lower than a random  $\varepsilon$ , the current point,  $P_i$  is changed by using the previous equations, in which the  $x$  and  $y$  indices should be different. Of course, the new  $P'_i$  is substituted by  $P_i$  if it has better value.

### 3. Proposed Dynamic Stochastic Fractal Search (DSFS)

As mentioned above, the SFS [30]–[32] has two important processes: the diffusion method and the updating strategy. Analyzing the diffusion method we know that a particle called seed originates, and others are generated and adhered to it, through diffusion that forms a chaotic-stochastic fractal branch, taking into account this fact the Stochastic Fractal Search (SFS) method was improved by adding diversity to the diffusion process for the particles in each iteration, in this way, the Gaussian random walk is helped, where the particles have more possibilities to exploit the search space and therefore do not stagnate in an optimal location. To control the diffusion parameter, a fuzzy inference system was introduced which dynamically adjusts the diffusion of the particles with a range of [0 1], this fuzzy system for control has two inputs (iteration and diversity) and one output (diffusion) as illustrated in Figure.1. In this case, Eqs. (11) and (12) represent diversity and iteration, respectively.

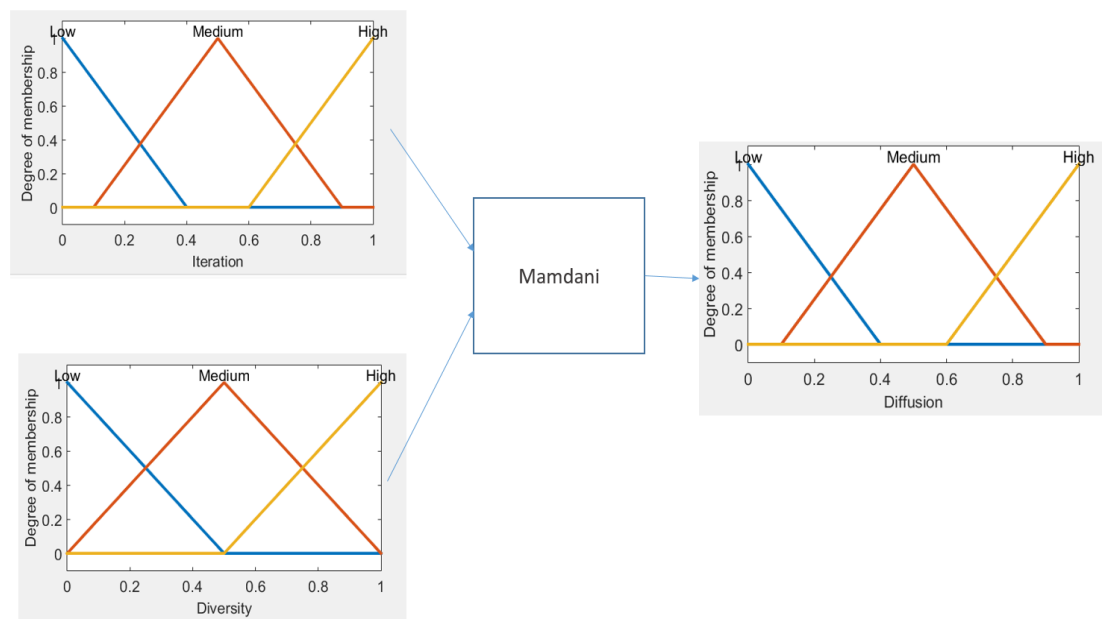


Figure 1. Type-1 fuzzy system for controlling of diffusion

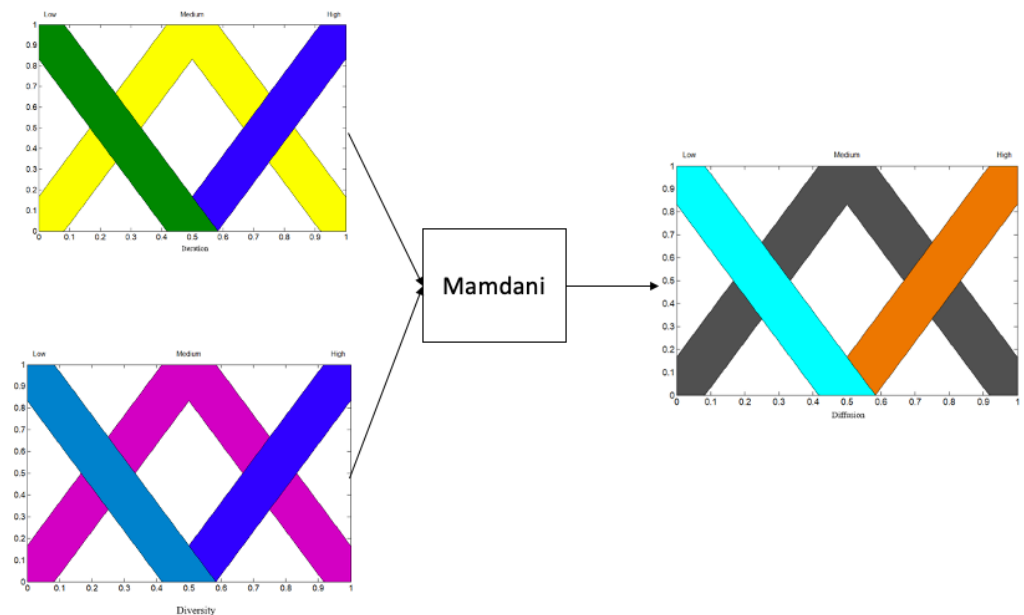


Figure 2. Type-2 fuzzy system for controlling the diffusion

### Fuzzy rules if-then

- If (Iteration is Low) and (Diversity is Low) then (Diffusion is High)
- If (Iteration is Low) and (Diversity is Medium) then (Diffusion is Medium)
- If (Iteration is Low) and (Diversity is High) then (Diffusion is Medium)
- If (Iteration is Medium) and (Diversity is Low) then (Diffusion is Medium)
- If (Iteration is Medium) and (Diversity is Medium) then (Diffusion is Medium)
- If (Iteration is Medium) and (Diversity is High) then (Diffusion is Medium)

If (Iteration is High) and (Diversity is Low) then (Diffusion is Medium)  
 If (Iteration is High) and (Diversity is Medium) then (Diffusion is Medium)  
 If (Iteration is High) and (Diversity is High) then (Diffusion is Low)

Eqs. 9 and 10 mathematically define the geometrical shape of the triangular functions, for type-1 and type-2 fuzzy logic, respectively [33]–[37].

$$\text{triangular}(u; a, b, c) = \begin{cases} 0, & u \leq a \\ \frac{u-a}{b-a}, & a \leq u \leq b \\ \frac{c-u}{c-b}, & b \leq u \leq c \\ 0, & c \leq u \end{cases}$$

(9)

$$\begin{aligned} \tilde{\mu}(x) &= [\underline{\mu}(x), \bar{\mu}(x)] = \\ \text{itrapatype2}(x, [a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, \alpha]) \\ \text{where } a_1 < a_2, b_1 < b_2, c_1 < c_2, d_1 < d_2 \\ \mu_1(x) &= \max\left(\min\left(\frac{x-a_1}{b_1-a_1}, 1, \frac{d_1-x}{d_1-c_1}\right), 0\right) \\ \mu_2(x) &= \max\left(\min\left(\frac{x-a_2}{b_2-a_2}, 1, \frac{d_2-x}{d_2-c_2}\right), 0\right) \\ \bar{\mu}(x) &= \begin{cases} \max(\mu_1(x), \mu_2(x)) & \forall x \notin (b_1, c_2) \\ 1 & \forall x \in (b_1, c_2) \end{cases} \\ \underline{\mu}(x) &= \min(\alpha, \min(\mu_1(x), \mu_2(x))) \end{aligned} \quad (10)$$

$$\text{Iteration} = \frac{\text{Current Iteration}}{\text{Total of Iterations}} \quad (11)$$

$$\text{Diversity}(S(t)) = \frac{1}{n} \sum_{x=1}^n \sqrt{\sum_{y=1}^D [P_x(t) - P_y(t)]^2} \quad (12)$$

Diversity contributes to the stochastic movement of the particles, to having more possibility of exploiting the entire search space, in each iteration it is dynamically adjusted close to the global optimum, thus the diffusion process is improved and therefore the most efficient method.

#### 4. Experimental results

To obtain an understanding of the effectiveness of the Dynamic Stochastic Fractal Search (DSFS) method, 30 functions of the CEC'2017 competition [38], summarized in Table 1, were evaluated. In Table 1 we can find several types of functions, such as:

unimodal, multimodal, hybrid and composite. To compare the optimization performance of the proposal respective to other methods, different number of dimensions (10, 30, 50 and 100) and different adaptation strategies for the parameters (using type-1 and type-2 fuzzy systems) were used.

s

Table 1. CEC’2017 optimization 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_Rastrigin Function	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

Composition Functions	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
	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

To evaluate the proposed method, the CEC'2017 benchmark mathematical functions that have been widely used in the literature [39], are considered in the tests. Table 2 shows at the top the dimensions used for each function, column 1 shows the function that is given as 1,2,3,4 ... 29., column 2 represented as f1 shows the global optimum (minimum) of each function, the next two columns contain the results obtained by the Dynamic Stochastic Fractal Search (DSFS) algorithm using type-1 fuzzy logic, the observable results are the average and standard deviation of each of the functions. Finally, columns 5 and 6 show the mean and standard deviations, respectively, of the functions when using type-2 fuzzy logic for the adaptation of the parameter values in DSFS.

Table 2. Dynamic Stochastic Fractal Search DSFS results with 10 dimensions

Dynamic Stochastic Fractal Search DSFS with 10 Dimensions					
Function	fi	Type 1 Fuzzy logic		Type 2 Fuzzy logic	
		Main	Std	Main	Std
f1	100	1.01E+02	4.87E-01	1.01E+02	7.85E-01
f2	200	2.00E+02	0.00E+00	2.00E+02	0.00E+00
f3	300	3.00E+02	3.73E-06	3.00E+02	5.04E-06
f4	400	4.01E+02	7.81E-01	4.00E+02	6.19E-01
f5	500	5.06E+02	2.04E+00	5.07E+02	2.64E+00



f6	600	6.00E+02	7.21E-08	6.00E+02	6.24E-08
f7	700	7.19E+02	3.45E+00	7.20E+02	3.32E+00
f8	800	8.07E+02	3.04E+00	8.07E+02	2.44E+00
f9	900	9.00E+02	0.00E+00	9.00E+02	0.00E+00
f10	1000	1.40E+03	1.70E+02	1.34E+03	1.31E+02
f11	1100	1.10E+03	9.31E-01	1.10E+03	1.00E+00
f12	1200	1.50E+03	1.04E+02	1.52E+03	1.02E+02
f13	1300	1.31E+03	4.03E+00	1.31E+03	4.14E+00
f14	1400	1.40E+03	1.69E+00	1.40E+03	2.01E+00
f15	1500	1.50E+03	7.24E-01	1.50E+03	8.88E-01
f16	1600	1.60E+03	2.79E-01	1.60E+03	3.64E-01
f17	1700	1.70E+03	2.68E+00	1.71E+03	4.41E+00
f18	1800	1.81E+03	2.56E+00	1.81E+03	2.57E+00
f19	1900	1.90E+03	4.15E-01	1.90E+03	4.45E-01
f20	2000	2.00E+03	1.39E-01	2.00E+03	4.73E-03
f21	2100	2.23E+03	5.08E+01	2.24E+03	5.40E+01
f22	2200	2.29E+03	2.17E+01	2.28E+03	3.93E+01
f23	2300	2.61E+03	2.80E+00	2.60E+03	5.61E+01
f24	2400	2.61E+03	1.22E+02	2.63E+03	1.14E+02
f25	2500	2.90E+03	1.08E+01	2.90E+03	1.15E+01
f26	2600	2.90E+03	2.15E-10	2.90E+03	3.02E-10
f27	2700	3.09E+03	1.97E+00	3.09E+03	2.14E+00
f28	2800	3.09E+03	4.20E+01	3.11E+03	5.69E+01
f29	2900	3.16E+03	1.01E+01	3.16E+03	9.65E+00
f30	3000	3.56E+03	2.30E+02	3.55E+03	1.19E+02

As can be seen, the results using type-1 and type-2 fuzzy systems did not have a significant visible difference using 10 dimensions, even so the results obtained were good on average, because most reached the optimal global of the functions. Table 3 has the same structure as the previous one, the difference is that these results were made with 30 dimensions, as it is 30 dimensions, the algorithm has a wider search space. In addition, the functions that are being optimized are complex, despite this situation, the method showed approaching the optimum of each function, and the values obtained with both types of fuzzy logic were also very close.

Table 3. Dynamic Stochastic Fractal Search DSFS results with 30 dimensions

Dynamic Stochastic Fractal Search DFSF with 30 Dimensions					
Type 1 Fuzzy logic				Type 2 Fuzzy logic	
Function	fi	Main	Std	Main	Std
f1	100	3.49E+03	2.76E+03	3.24E+03	2.74E+03
f2	200	3.06E+16	7.11E+16	7.59E+17	3.98E+18
f3	300	8.40E+03	3.95E+03	8.79E+03	4.60E+03

f4	400	<b>4.87E+02</b>	3.56E+01	<b>4.80E+02</b>	3.39E+01
f5	500	6.11E+02	2.27E+01	6.00E+02	2.02E+01
f6	600	<b>6.00E+02</b>	1.36E-02	<b>6.00E+02</b>	1.13E-02
f7	700	8.53E+02	1.68E+01	8.60E+02	1.70E+01
f8	800	9.05E+02	2.12E+01	9.06E+02	2.27E+01
f9	900	<b>9.01E+02</b>	6.89E-01	<b>9.04E+02</b>	1.44E+01
f10	1000	6.21E+03	6.89E-01	6.05E+03	5.03E+02
f11	1100	<b>1.19E+03</b>	2.31E+01	<b>1.19E+03</b>	2.90E+01
f12	1200	1.56E+05	1.03E+05	1.81E+05	1.48E+05
f13	1300	3.56E+03	9.09E+02	4.01E+03	1.00E+03
f14	1400	1.50E+03	1.03E+01	1.50E+03	1.06E+01
f15	1500	1.70E+03	4.43E+01	1.70E+03	3.95E+01
f16	1600	2.45E+03	2.66E+02	2.47E+03	2.14E+02
f17	1700	1.85E+03	6.65E+01	1.86E+03	8.06E+01
f18	1800	2.87E+03	6.86E+02	2.69E+03	3.69E+02
f19	1900	<b>1.99E+03</b>	1.88E+01	<b>1.99E+03</b>	1.87E+01
f20	2000	2.39E+03	2.58E+01	<b>2.22E+03</b>	1.12E+02
f21	2100	2.40E+03	2.28E+01	2.39E+03	2.51E+01
f22	2200	2.30E+03	1.50E-02	2.30E+03	7.44E-03
f23	2300	2.74E+03	2.86E+01	2.73E+03	2.26E+01
f24	2400	2.91E+03	3.37E+01	2.90E+03	3.04E+01
f25	2500	2.89E+03	1.73E+00	2.89E+03	9.76E-01
f26	2600	4.28E+03	5.50E+02	4.22E+03	6.59E+02
f27	2700	3.22E+03	8.08E+00	3.22E+03	7.37E+00
f28	2800	3.21E+03	1.25E+01	3.21E+03	1.27E+01
f29	2900	3.63E+03	1.06E+02	3.59E+03	1.13E+02
f30	3000	1.32E+04	3.29E+03	1.51E+04	5.55E+03

Table 4. Dynamic Stochastic Fractal Search DSFS results with 50 dimensions

Dynamic Stochastic Fractal Search DSFS with 50 Dimensions					
Function	fi	Type 1 Fuzzy logic		Type 2 Fuzzy logic	
		Main	Std	Main	Std
f1	100	9.00E+04	4.30E+04	8.42E+04	6.69E+04
f2	200	1.20E+40	4.53E+40	5.08E+39	1.52E+40
f3	300	6.01E+04	9.72E+03	6.28E+04	1.39E+04
f4	400	<b>5.72E+02</b>	4.05E+01	<b>5.69E+02</b>	4.26E+01
f5	500	<b>7.68E+02</b>	4.35E+01	<b>7.69E+02</b>	4.59E+01
f6	600	<b>6.01E+02</b>	1.35E-01	<b>6.01E+02</b>	1.55E-01
f7	700	1.05E+03	3.18E+01	1.06E+03	2.61E+01
f8	800	1.06E+03	3.69E+01	1.06E+03	4.42E+01
f9	900	1.13E+03	1.49E+02	1.14E+03	1.22E+02
f10	1000	1.13E+04	4.84E+02	1.12E+04	6.53E+02

f11	1100	<b>1.36E+03</b>	3.00E+01	<b>1.36E+03</b>	4.30E+01
f12	1200	3.54E+06	1.42E+06	3.47E+06	1.96E+06
f13	1300	1.48E+04	1.17E+04	1.67E+04	1.36E+04
f14	1400	<b>1.50E+03</b>	1.03E+01	1.80E+03	8.36E+01
f15	1500	4.14E+03	1.54E+03	3.75E+03	8.43E+02
f16	1600	3.52E+03	3.87E+02	3.58E+03	4.80E+02
f17	1700	3.05E+03	2.37E+02	3.04E+03	2.56E+02
f18	1800	5.42E+04	3.34E+04	5.39E+04	3.20E+04
f19	1900	7.10E+03	4.05E+03	7.73E+03	5.45E+03
f20	2000	3.10E+03	2.16E+02	3.14E+03	2.36E+02
f21	2100	<b>2.55E+03</b>	4.55E+01	<b>2.55E+03</b>	4.99E+01
f22	2200	1.10E+04	4.12E+03	1.23E+04	3.40E+03
f23	2300	3.00E+03	4.43E+01	2.98E+03	4.91E+01
f24	2400	3.14E+03	5.98E+01	3.14E+03	5.85E+01
f25	2500	3.07E+03	2.52E+01	3.08E+03	2.22E+01
f26	2600	6.07E+03	5.92E+02	6.16E+03	4.77E+02
f27	2700	3.41E+03	4.40E+01	3.40E+03	4.09E+01
f28	2800	3.36E+03	3.36E+01	3.35E+03	3.44E+01
f29	2900	4.15E+03	2.71E+02	4.17E+03	2.55E+02
f30	3000	3.25E+06	6.84E+05	3.24E+06	8.20E+05

Table 5. Dynamic Stochastic Fractal Search DSFS results with 100 dimensions

Dynamic Stochastic Fractal Search DSFS with 100 Dimensions					
Type-1 Fuzzy logic				Type-2 Fuzzy logic	
Function	fi	Main	Std	Main	Std
f1	100	1.15E+08	4.36E+07	1.08E+08	3.31E+07
f2	200	1.34E+108	8.85E+108	1.14E+111	5.81E+111
f3	300	2.53E+05	2.78E+04	2.56E+05	2.78E+04
f4	400	<b>9.23E+02</b>	4.74E+01	<b>9.37E+02</b>	6.10E+01
f5	500	1.29E+03	9.15E+01	1.29E+03	7.11E+01
f6	600	<b>6.07E+02</b>	1.33E+00	<b>6.07E+02</b>	1.02E+00
f7	700	1.72E+03	5.61E+01	1.73E+03	5.30E+01
f8	800	1.58E+03	7.70E+01	1.59E+03	8.69E+01
f9	900	1.19E+04	3.07E+03	1.28E+04	4.21E+03
f10	1000	2.71E+04	1.04E+03	2.72E+04	9.91E+02
f11	1100	1.62E+04	3.86E+03	1.63E+04	4.35E+03
f12	1200	6.66E+07	2.09E+07	7.05E+07	1.74E+07
f13	1300	5.66E+03	1.80E+03	6.16E+03	2.89E+03
f14	1400	3.36E+05	2.50E+05	2.36E+05	1.44E+05
f15	1500	4.46E+03	4.66E+03	4.39E+03	3.13E+03
f16	1600	7.97E+03	8.58E+02	7.70E+03	7.93E+02

---

f17	1700	6.04E+03	3.39E+02	5.90E+03	5.76E+02
f18	1800	5.97E+05	3.86E+05	6.45E+05	3.99E+05
f19	1900	3.60E+03	1.65E+03	3.49E+03	1.56E+03
f20	2000	<b>2.18E+03</b>	8.28E+01	6.27E+03	4.30E+02
f21	2100	3.11E+03	6.75E+01	3.11E+03	6.06E+01
f22	2200	2.96E+04	8.67E+02	2.97E+04	8.23E+02
f23	2300	3.59E+03	5.69E+01	3.57E+03	7.87E+01
f24	2400	4.09E+03	9.76E+01	4.08E+03	1.19E+02
f25	2500	3.63E+03	6.17E+01	3.63E+03	5.46E+01
f26	2600	1.41E+04	1.04E+03	1.42E+04	8.28E+02
f27	2700	3.72E+03	7.06E+01	3.73E+03	5.89E+01
f28	2800	3.96E+03	1.11E+02	3.96E+03	1.35E+02
f29	2900	7.96E+03	5.34E+02	7.96E+03	5.11E+02
f30	3000	3.04E+05	1.27E+05	3.18E+05	1.68E+05

---

Multimodal functions are more difficult to optimize than unimodal ones due to the complexity that they represent, because the algorithms must escape or avoid local optima and arrive to the global optimal solution. In this study, not only are unimodal and multimodal functions being optimized, but also hybrid and complex functions are being optimized using different values of dimensions, as can be seen in Tables 4 and 5 with 50 and 100 dimensions respectively. In addition, the values obtained with the variants using Type-1 and Type-2 fuzzy systems for adaptation of parameters, show that the method had some degree of difficulty in reaching the global optimum, even so, there provide good approximation values showing that the improved method is efficient in optimization tasks.

## 5. Discussion of Results

In the literature, we can find the hybrid firefly and particle swarm optimization algorithm for solving expensive computationally problems [40] that was used in the optimization the CEC'2017 functions and its performance was compared to other 5 optimization methods, namely: the combination the Hybrid firefly algorithm and particles swarm optimization (HFPSO), also Firefly Algorithm (FA), Particle Swarm Optimization (PSO), Hybrid PSO and Firefly Algorithm (HPSOFF) and Hybrid Firefly and PSO (FFPSO). For this reason, we decided to consider it as a reference for comparison to test the efficacy of the Dynamic Stochastic Fractal Search (DFSf) method. The experimentation was carried out with the following specifications: 20 independent runs, for each case, the maximum number of evaluations of 500 was used for 10 dimensions (10D) and 1500 for 30 dimensions problems (30D). Tables 6 and 7 show the results with 10 dimensions and 30 dimensions respectively, of this comparison study.

Table 6. CEC’2017 10 dimensions results summary [40]

CEC 2017 10D results summary.

Fn.	PSO		FA		FFPSO		HPSOFF		HFPSO	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
1	3,2165E+08	2,3861E+08	1,6258E+09	1,2694E+09	2,2812E+10	7,5101E+09	<b>1,7345E+08</b>	2,8025E+08	2,4541E+08	9,8060E+08
2	4,6348E+08	7,5312E+08	3,3699E+11	1,4964E+12	1,8124E+16	3,4626E+16	1,8320E+08	3,2778E+08	<b>1,4746E+08</b>	4,9126E+08
3	1,7908E+04	4,9711E+03	3,0383E+04	2,0969E+04	8,9010E+07	1,8367E+08	2,2767E+04	1,0668E+04	<b>1,1744E+04</b>	5,9568E+03
4	5,4787E+02	3,2899E+02	5,1021E+02	5,6927E+01	3,3173E+03	1,6661E+03	4,5895E+02	4,3844E+01	<b>4,4101E+02</b>	4,5452E+01
5	5,7248E+02	1,6763E+01	5,6142E+02	1,8732E+01	6,7964E+02	2,4753E+01	5,6661E+02	1,7223E+01	<b>5,5292E+02</b>	1,8402E+01
6	6,3032E+02	1,5130E+01	<b>6,1687E+02</b>	5,2386E+00	6,8447E+02	1,7108E+01	6,2494E+02	9,6435E+00	6,2194E+02	1,3521E+01
7	7,8146E+02	1,1902E+01	7,9202E+02	2,2557E+01	1,0146E+03	6,1461E+01	7,8107E+02	1,5138E+01	<b>7,6343E+02</b>	1,7263E+01
8	8,6055E+02	9,0266E+00	8,5822E+02	1,8953E+01	9,3140E+02	2,0012E+01	8,5221E+02	1,3544E+01	<b>8,3952E+02</b>	1,4438E+01
9	<b>1,1099E+03</b>	2,2269E+02	1,1843E+03	1,7125E+02	3,6784E+03	8,9314E+02	1,2238E+03	2,7506E+02	1,1775E+03	3,0667E+02
10	3,0080E+03	2,9923E+02	2,6285E+03	3,2258E+02	3,8614E+03	2,8731E+02	2,6270E+03	3,7715E+02	<b>2,2670E+03</b>	3,7933E+02
11	1,6759E+03	9,9140E+02	3,5406E+03	4,9132E+03	6,4221E+04	1,6948E+05	1,2625E+03	1,2696E+02	<b>1,1959E+03</b>	5,2380E+01
12	1,8611E+07	8,8766E+06	7,7087E+06	6,2420E+06	3,0343E+09	1,5253E+09	<b>2,8243E+06</b>	3,0772E+06	2,8998E+06	4,1307E+06
13	4,9528E+05	3,9492E+05	1,8233E+04	2,0474E+04	6,5693E+08	6,4239E+08	1,4573E+04	7,4206E+03	<b>1,1835E+04</b>	7,6845E+03
14	7,6428E+03	7,0146E+03	1,0411E+04	9,2273E+03	8,5002E+05	1,0892E+06	5,0414E+03	3,4439E+03	<b>4,6478E+03</b>	4,1776E+03
15	4,8266E+04	4,6631E+04	1,8770E+04	1,3651E+04	5,9980E+07	9,8559E+07	<b>1,3609E+04</b>	1,1050E+04	2,2845E+04	2,3650E+04
16	2,1201E+03	1,9609E+02	1,9873E+03	1,5647E+02	2,8816E+03	3,1814E+02	2,0516E+03	1,5269E+02	<b>1,9629E+03</b>	1,5877E+02
17	1,8650E+03	5,9215E+01	<b>1,8035E+03</b>	5,1924E+01	2,4780E+03	2,7983E+02	1,8217E+03	4,6836E+01	1,8248E+03	8,3997E+01
18	7,0533E+05	8,2307E+05	<b>2,1706E+04</b>	1,6198E+04	2,0293E+09	1,8875E+09	8,1675E+04	1,9730E+05	2,9732E+04	1,7941E+04
19	3,5868E+04	2,5999E+04	<b>1,4600E+04</b>	1,2364E+04	2,4345E+07	4,1842E+07	1,8030E+04	9,0940E+03	3,5177E+04	3,8300E+04
20	2,2668E+03	1,1628E+02	<b>2,1498E+03</b>	5,6944E+01	2,5686E+03	1,6145E+02	2,1834E+03	9,3493E+01	2,2023E+03	3,7933E+02
21	2,3553E+03	3,7009E+01	2,3532E+03	1,0085E+01	2,4638E+03	3,5194E+01	2,3488E+03	4,6622E+01	<b>2,3377E+03</b>	4,7750E+01
22	2,6684E+03	7,7654E+02	2,4297E+03	1,9109E+02	4,3663E+03	6,4636E+02	<b>2,4196E+03</b>	2,0260E+02	2,5975E+03	5,8755E+02
23	2,6906E+03	3,4234E+01	<b>2,6656E+03</b>	1,2682E+01	3,0293E+03	2,0917E+02	2,6783E+03	2,3378E+01	2,6757E+03	2,8668E+01
24	2,7972E+03	9,4842E+01	2,7778E+03	6,4001E+01	3,1865E+03	1,5208E+02	2,7337E+03	1,1168E+02	<b>2,7128E+03</b>	1,4672E+02
25	2,9648E+03	1,5755E+02	2,9868E+03	4,3648E+01	4,4005E+03	5,4274E+02	<b>2,9515E+03</b>	1,8425E+01	2,9581E+01	5,0193E+01
26	3,4545E+03	6,3256E+02	3,5102E+03	4,3450E+02	5,0797E+03	4,2029E+02	3,4471E+03	5,3585E+02	<b>3,2147E+03</b>	3,4227E+02
27	3,1529E+03	3,9793E+01	<b>3,1221E+03</b>	1,9657E+01	3,7507E+03	2,6732E+02	3,1356E+03	3,3564E+01	3,1538E+03	3,9370E+01
28	<b>3,3450E+03</b>	1,0498E+02	3,4768E+03	1,8028E+02	4,3603E+03	2,8037E+02	3,4064E+03	9,4585E+01	3,4422E+03	1,0794E+02
29	3,4299E+03	8,8769E+01	<b>3,3299E+03</b>	6,6574E+01	4,1052E+03	3,2667E+02	3,3721E+03	1,0183E+02	3,3864E+03	9,3976E+01
30	4,1456E+06	4,2666E+06	<b>2,6353E+06</b>	2,6696E+06	3,0366E+08	2,4690E+08	3,6005E+06	2,9950E+06	4,2723E+06	3,7549E+06

Table 7. CEC’2017 30 dimensions results summary [40].

CEC 2017 30D results summary.

Fn.	PSO		FA		FFPSO		HPSOFF		HFPSO	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
1	5,5573E+09	3,3893E+09	3,3061E+10	4,4031E+09	1,1348E+11	1,2767E+10	6,0544E+09	2,1716E+09	<b>1,4877E+09</b>	8,9773E+08
2	3,9099E+32	1,6788E+33	1,8379E+39	3,6984E+39	1,0418E+57	4,6498E+57	3,1775E+34	1,0461E+35	<b>4,2895E+30</b>	1,3632E+31
3	1,3257E+05	3,4533E+04	2,1556E+05	4,4961E+04	6,7723E+09	1,9830E+10	1,3635E+05	3,6125E+04	<b>1,3159E+05</b>	5,0828E+04
4	9,0860E+02	1,7332E+02	6,8240E+03	1,7720E+03	4,4924E+04	1,0769E+04	1,2583E+03	3,3792E+02	<b>6,8099E+02</b>	7,2977E+01
5	8,0387E+02	2,0464E+01	8,7976E+02	2,1895E+01	1,1284E+03	6,7520E+01	8,1363E+02	3,4695E+01	<b>7,4431E+02</b>	2,8310E+01
6	6,5628E+02	1,3442E+01	6,7297E+02	6,4935E+00	7,2989E+02	1,2755E+01	6,7252E+02	8,5569E+00	<b>6,5458E+02</b>	1,4862E+01
7	1,1273E+03	3,0590E+01	1,7985E+03	1,6329E+02	2,9927E+03	2,0330E+02	1,1241E+03	5,3149E+01	<b>1,0634E+03</b>	3,8217E+01
8	1,0857E+03	2,1442E+01	1,1591E+03	3,0228E+01	1,3608E+03	3,8288E+01	1,0756E+03	3,0109E+01	<b>1,0174E+03</b>	3,4938E+01
9	<b>6,6038E+03</b>	2,5645E+03	1,3053E+04	1,6887E+03	3,0274E+04	3,4821E+03	9,0601E+03	3,0384E+03	9,0430E+03	2,4238E+03
10	9,2920E+03	3,1888E+02	9,4664E+03	4,8533E+02	1,0715E+04	4,5678E+02	8,9099E+03	5,0987E+02	<b>7,4962E+03</b>	9,1080E+02
11	3,6404E+03	9,2311E+02	1,1963E+04	3,7161E+03	3,4403E+04	1,4283E+04	4,5469E+03	1,0526E+03	<b>2,2880E+03</b>	6,8090E+02
12	6,4666E+08	3,0524E+08	3,0340E+09	9,0341E+08	2,7727E+10	5,6036E+09	3,3040E+08	2,2407E+08	<b>5,9479E+07</b>	3,7560E+07
13	1,9336E+08	5,8827E+07	1,1545E+09	6,1763E+08	2,7922E+10	1,2029E+10	2,7671E+07	1,9963E+07	<b>5,0687E+06</b>	1,5852E+07
14	1,1117E+06	7,8687E+05	1,8933E+06	1,8783E+06	5,6035E+07	3,5343E+07	1,1120E+06	1,1345E+06	<b>6,6292E+05</b>	9,6993E+05
15	3,9118E+07	1,8225E+07	1,0725E+08	6,1741E+07	7,1411E+09	3,1801E+09	1,5676E+06	1,1456E+06	<b>7,3512E+04</b>	2,3482E+04
16	3,9514E+03	4,2128E+02	4,4164E+03	2,9240E+02	9,6352E+03	2,6750E+03	3,6383E+03	4,0419E+02	<b>3,1951E+03</b>	4,2601E+02
17	2,6556E+03	2,3445E+02	3,0044E+03	1,8143E+02	3,3347E+04	4,9223E+04	2,5254E+03	2,5071E+02	<b>2,2984E+02</b>	2,1441E+02
18	7,9043E+06	6,2705E+06	2,9281E+07	1,9063E+07	1,0554E+09	7,7432E+08	8,5589E+06	9,7364E+06	<b>2,5984E+06</b>	2,1944E+06
19	5,1700E+07	2,9419E+07	1,6613E+08	6,2094E+07	8,2628E+09	3,4065E+09	8,4341E+06	9,5782E+06	<b>5,9487E+05</b>	5,7691E+05
20	3,0177E+03	2,1693E+02	3,1126E+03	1,3585E+02	3,7975E+03	2,1617E+02	2,9083E+03	2,5178E+02	<b>2,7685E+03</b>	1,8760E+02
21	2,6009E+03	2,2044E+01	2,6405E+03	2,6874E+01	2,8880E+03	6,5735E+01	2,5982E+03	5,7252E+01	<b>2,5087E+03</b>	2,9204E+01
22	6,8425E+03	3,8923E+03	8,0003E+03	1,8365E+03	1,2117E+04	3,2675E+02	<b>5,7433E+03</b>	3,1589E+03	5,7966E+03	3,2553E+03
23	3,0819E+03	1,0238E+02	3,1542E+03	4,9848E+01	4,0749E+03	2,8961E+02	3,0969E+03	9,0742E+01	<b>2,9597E+03</b>	7,4124E+01
24	3,2139E+03	7,7916E+01	3,3056E+03	5,1836E+01	4,4529E+03	2,5580E+02	3,2719E+03	9,8133E+01	<b>3,1742E+03</b>	1,0577E+02
25	3,2145E+03	6,7870E+01	4,9480E+03	4,0432E+02	1,5958E+04	3,1469E+03	3,2320E+03	1,4716E+02	<b>3,0623E+03</b>	7,0241E+01
26	6,6461E+03	1,1813E+03	8,8041E+03	6,0012E+02	1,6163E+04	1,5021E+03	7,2386E+03	1,6475E+03	<b>5,6590E+03</b>	1,5232E+03
27	3,3573E+03	8,5636E+01	3,6059E+03	6,7467E+01	5,8099E+03	7,3079E+02	3,4360E+03	9,9726E+01	<b>3,3243E+03</b>	6,1360E+01
28	3,6388E+03	1,2870E+02	5,6940E+03	4,5852E+02	1,3535E+04	1,8714E+03	3,7938E+03	2,4230E+02	<b>3,3925E+03</b>	5,6285E+01
29	4,9240E+03	3,1984E+02	5,7051E+03	3,5852E+02	9,3924E+04	1,0965E+05	4,8406E+03	4,0875E+02	<b>4,4249E+03</b>	2,5350E+02
30	4,8514E+07	1,7476E+07	1,7163E+08	6,1288E+07	4,1425E+09	1,5608E+09	2,5749E+07	1,9168E+07	<b>4,6082E+06</b>	2,7679E+06

Table 8. HFPSO vs DSFS results with 10 dimensions

Function	HFPSO [40]				DSFS		DSFS	
	fi		Mean		Type 1 fuzzy logic		Type 2 fuzzy logic	
					Mean	Std	Mean	Std

f1	100	9.81E+08	1.01E+02	<b>1.01E+02</b>	4.87E-01	<b>1.01E+02</b>	7.85E-01
f2	200	4.91E+08	2.00E+02	<b>2.00E+02</b>	0.00E+00	<b>2.00E+02</b>	0.00E+00
f3	300	5.96E+03	3.00E+02	<b>3.00E+02</b>	3.73E-06	<b>3.00E+02</b>	5.04E-06
f4	400	4.55E+01	4.01E+02	<b>4.01E+02</b>	7.81E-01	<b>4.00E+02</b>	6.19E-01
f5	500	1.84E+01	5.06E+02	<b>5.06E+02</b>	2.04E+00	<b>5.07E+02</b>	2.64E+00
f6	600	1.35E+01	6.00E+02	<b>6.00E+02</b>	7.21E-08	<b>6.00E+02</b>	6.24E-08
f7	700	1.73E+01	7.19E+02	<b>7.19E+02</b>	3.45E+00	<b>7.20E+02</b>	3.32E+00
f8	800	1.44E+01	8.07E+02	<b>8.07E+02</b>	3.04E+00	<b>8.07E+02</b>	2.44E+00
f9	900	3.07E+02	9.00E+02	<b>9.00E+02</b>	0.00E+00	<b>9.00E+02</b>	0.00E+00
f10	1000	3.79E+02	1.40E+03	<b>1.40E+03</b>	1.70E+02	<b>1.34E+03</b>	1.31E+02
f11	1100	5.24E+01	1.10E+03	<b>1.10E+03</b>	9.31E-01	<b>1.10E+03</b>	1.00E+00
f12	1200	4.13E+06	1.50E+03	<b>1.50E+03</b>	1.04E+02	<b>1.52E+03</b>	1.02E+02
f13	1300	7.68E+03	1.31E+03	<b>1.31E+03</b>	4.03E+00	<b>1.31E+03</b>	4.14E+00
f14	1400	4.18E+03	1.40E+03	<b>1.40E+03</b>	1.69E+00	<b>1.40E+03</b>	2.01E+00
f15	1500	2.37E+04	1.50E+03	<b>1.50E+03</b>	7.24E-01	<b>1.50E+03</b>	8.88E-01
f16	1600	1.59E+02	1.60E+03	<b>1.60E+03</b>	2.79E-01	<b>1.60E+03</b>	3.64E-01
f17	1700	8.40E+01	1.70E+03	<b>1.70E+03</b>	2.68E+00	<b>1.71E+03</b>	4.41E+00
f18	1800	1.79E+04	1.81E+03	<b>1.81E+03</b>	2.56E+00	<b>1.81E+03</b>	2.57E+00
f19	1900	3.83E+04	1.90E+03	<b>1.90E+03</b>	4.15E-01	<b>1.90E+03</b>	4.45E-01
f20	2000	1.08E+02	2.00E+03	<b>2.00E+03</b>	1.39E-01	<b>2.00E+03</b>	4.73E-03
f21	2100	4.78E+01	2.23E+03	<b>2.23E+03</b>	5.08E+01	<b>2.24E+03</b>	5.40E+01
f22	2200	5.88E+02	2.29E+03	<b>2.29E+03</b>	2.17E+01	<b>2.28E+03</b>	3.93E+01
f23	2300	2.87E+01	2.61E+03	<b>2.61E+03</b>	2.80E+00	<b>2.60E+03</b>	5.61E+01
f24	2400	1.47E+02	2.61E+03	<b>2.61E+03</b>	1.22E+02	<b>2.63E+03</b>	1.14E+02
f25	2500	5.02E+01	2.90E+03	<b>2.90E+03</b>	1.08E+01	<b>2.90E+03</b>	1.15E+01
f26	2600	3.42E+02	2.90E+03	<b>2.90E+03</b>	2.15E-10	<b>2.90E+03</b>	3.02E-10
f27	2700	3.94E+01	3.09E+03	<b>3.09E+03</b>	1.97E+00	<b>3.09E+03</b>	2.14E+00
f28	2800	1.08E+02	3.09E+03	<b>3.09E+03</b>	4.20E+01	<b>3.11E+03</b>	5.69E+01
f29	2900	9.40E+01	3.16E+03	<b>3.16E+03</b>	1.01E+01	<b>3.16E+03</b>	9.65E+00
f30	3000	3.75E+06	3.56E+03	<b>3.56E+03</b>	2.30E+02	<b>3.55E+03</b>	1.19E+02

Table 9. HFPSO vs DSFS results with 30 dimensions

HFPSO [40]				DSFS		DSFS	
				Type 1 fuzzy logic		Type 2 fuzzy logic	
Function	fi	Mean	Std	Mean	Std	Mean	Std
f1	100	9.81E+08	1.01E+02	3.49E+03	2.76E+03	3.24E+03	2.74E+03
f2	200	4.91E+08	2.00E+02	3.06E+16	7.11E+16	7.59E+17	3.98E+18
f3	300	5.96E+03	3.00E+02	8.40E+03	3.95E+03	8.79E+03	4.60E+03
f4	400	4.55E+01	4.01E+02	4.87E+02	3.56E+01	4.80E+02	3.39E+01
f5	500	1.84E+01	5.06E+02	6.11E+02	2.27E+01	6.00E+02	2.02E+01



---

f6	600	1.35E+01	6.00E+02	6.00E+02	1.36E-02	6.00E+02	1.13E-02
f7	700	1.73E+01	7.19E+02	8.53E+02	1.68E+01	8.60E+02	1.70E+01
f8	800	1.44E+01	8.07E+02	9.05E+02	2.12E+01	9.06E+02	2.27E+01
f9	900	3.07E+02	9.00E+02	9.01E+02	6.89E-01	9.04E+02	1.44E+01
f10	1000	3.79E+02	1.40E+03	6.21E+03	6.89E-01	6.05E+03	5.03E+02
f11	1100	5.24E+01	1.10E+03	1.19E+03	2.31E+01	1.19E+03	2.90E+01
f12	1200	4.13E+06	1.50E+03	1.56E+05	1.03E+05	1.81E+05	1.48E+05
f13	1300	7.68E+03	1.31E+03	3.56E+03	9.09E+02	4.01E+03	1.00E+03
f14	1400	4.18E+03	1.40E+03	1.50E+03	1.03E+01	1.50E+03	1.06E+01
f15	1500	2.37E+04	1.50E+03	1.70E+03	4.43E+01	1.70E+03	3.95E+01
f16	1600	1.59E+02	1.60E+03	2.45E+03	2.66E+02	2.47E+03	2.14E+02
f17	1700	8.40E+01	1.70E+03	1.85E+03	6.65E+01	1.86E+03	8.06E+01
f18	1800	1.79E+04	1.81E+03	2.87E+03	6.86E+02	2.69E+03	3.69E+02
f19	1900	3.83E+04	1.90E+03	1.99E+03	1.88E+01	1.99E+03	1.87E+01
f20	2000	1.08E+02	2.00E+03	2.39E+03	2.58E+01	2.22E+03	1.12E+02
f21	2100	4.78E+01	2.23E+03	2.40E+03	2.28E+01	2.39E+03	2.51E+01
f22	2200	5.88E+02	2.29E+03	2.30E+03	1.50E-02	2.30E+03	7.44E-03
f23	2300	2.87E+01	2.61E+03	2.74E+03	2.86E+01	2.73E+03	2.26E+01
f24	2400	1.47E+02	2.61E+03	2.91E+03	3.37E+01	2.90E+03	3.04E+01
f25	2500	5.02E+01	2.90E+03	2.89E+03	1.73E+00	2.89E+03	9.76E-01
f26	2600	3.42E+02	2.90E+03	4.28E+03	5.50E+02	4.22E+03	6.59E+02
f27	2700	3.94E+01	3.09E+03	3.22E+03	8.08E+00	3.22E+03	7.37E+00
f28	2800	1.08E+02	3.09E+03	3.21E+03	1.25E+01	3.21E+03	1.27E+01
f29	2900	9.40E+01	3.16E+03	3.63E+03	1.06E+02	3.59E+03	1.13E+02
f30	3000	3.75E+06	3.56E+03	1.32E+04	3.29E+03	1.51E+04	5.55E+03

---

The combination of Firefly Algorithm and Particle Swarm Optimization (HFPSO) was the one that generated the best results in the comparison that was made in the reference article with respect to the other 5 optimization algorithms. For this reason, in Tables 8 and 9 only the results of the HFPSO are compared against the proposed DSFS with both types of fuzzy logic. The experimentation of the CEC'2017 functions with 10D and 30 D respectively, show us that the Dynamic Stochastic Fractal Search (DSFS) method obtained on average better results in finding the global solutions of the functions. As previously mentioned, the results of Dynamic Stochastic Fractal Search (DSFS) with type-1 and type-2 fuzzy systems were very close, therefore, in comparison with HFPSO, better overall results were also obtained for each function. Figures 3 and 4 graphically illustrate a comparison of the proposed DSFS against the hybrid HFPSO for 10 and 30 dimensions, respectively.

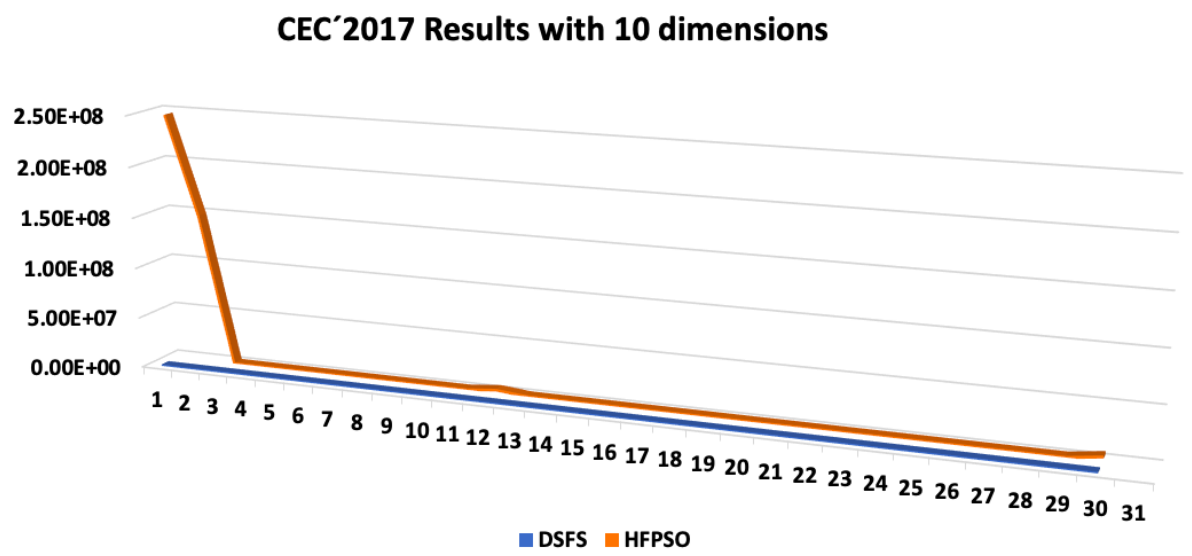


Figure 3. DSFS vs HFPSO Results with 10 dimensions.

On the x-axis we can find the values obtained with the DSFS and HFPSO methods, and the horizontal axis shows the average results of the evaluated functions. The DSFS approach, indicated by the blue sample line, has a better efficiency than the HFPSO orange line, because the values are closer to the global optima, as seen in Figure 3.

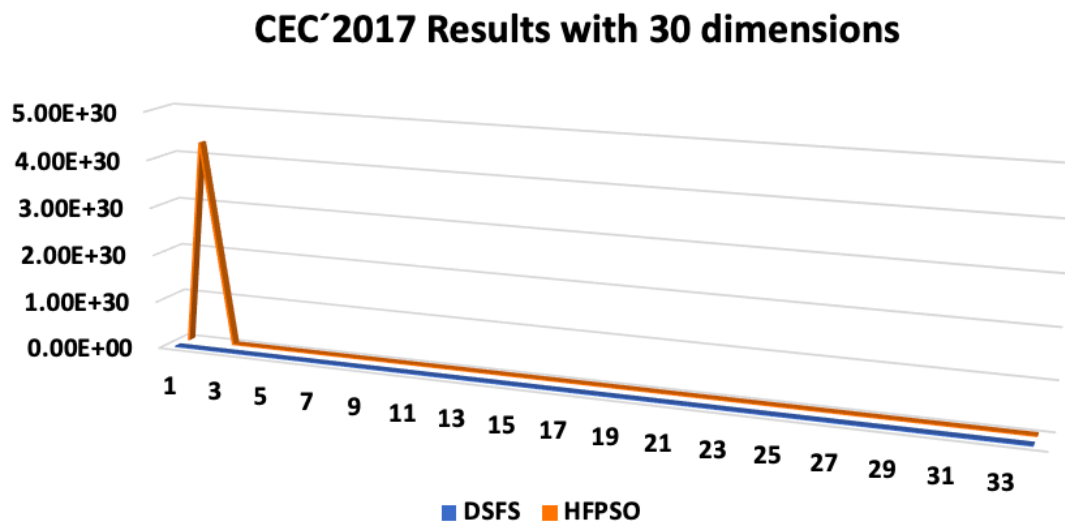


Figure 4. Results of CEC'2017 with 30 dimensions.

Figure 4 illustrates the results obtained for the DSFS and HFPSO, using 30 dimensions, the vertical axis contains the values reached by the methods when evaluating the CEC'2017 functions and the horizontal axis shows the corresponding function. The orange line is the representative of the HFPSO algorithm where it is observed that in the first functions the values were very separate from the optimal ones, then for the f1, f2, f3,



f4 functions the achieved performance can be viewed as unsatisfactory, with results up to  $10^{+30}$ . On the other hand, the DFSF method obtained results closer to the global optimal being represented by the blue line, we will have to remember that the dimensions are high and therefore, the performance of the methods is not the same, to use lower dimensions where the efficiency of the algorithms is much better for each of the functions.

To determine which of HFPSO or DSFS provided the closest to optimal result for each function, a statistical comparison was performed using the parametric z test. The formula for the Z test is expressed mathematically in the following fashion:

$$Z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sigma_{\bar{x}_1 - \bar{x}_2}} \quad (13)$$

where:

$\bar{x}_1 - \bar{x}_2$ : It is the observed difference in the mean values of the methods.

$\mu_1 - \mu_2$ : It is the expected difference in the mean values of the methods.

$\sigma_{\bar{x}_1 - \bar{x}_2}$ : Standard error of the differences.

As can be seen in Table 10, column 7, the values obtained by the z-test, provide statistical evidence that the DSFS method using type-1 fuzzy logic is significantly better than HFPSO, in functions with 10 dimensions (Bold indicates best values).

Table 10. Z-test for 10 dimensions.

Function	fi	HFPSO [40]		DSFS Type 1 fuzzy logic		z
		Mean	Std	Mean	Std	
f1	100	9.81E+08	1.01E+02	<b>1.01E+02</b>	4.87E-01	3.76E+07
f2	200	4.91E+08	2.00E+02	<b>2.00E+02</b>	0.00E+00	9.51E+06
f3	300	5.96E+03	3.00E+02	<b>3.00E+02</b>	3.73E-06	7.31E+01
f4	400	4.55E+01	4.01E+02	<b>4.01E+02</b>	7.81E-01	-3.43E+00
f5	500	1.84E+01	5.06E+02	<b>5.06E+02</b>	2.04E+00	-3.73E+00
f6	600	1.35E+01	6.00E+02	<b>6.00E+02</b>	7.21E-08	-3.79E+00
f7	700	1.73E+01	7.19E+02	<b>7.19E+02</b>	3.45E+00	-3.78E+00
f8	800	1.44E+01	8.07E+02	<b>8.07E+02</b>	3.04E+00	-3.80E+00
f9	900	3.07E+02	9.00E+02	<b>9.00E+02</b>	0.00E+00	-2.55E+00
f10	1000	3.79E+02	1.40E+03	<b>1.40E+03</b>	1.70E+02	-2.82E+00
f11	1100	5.24E+01	1.10E+03	<b>1.10E+03</b>	9.31E-01	-3.69E+00
f12	1200	4.13E+06	1.50E+03	<b>1.50E+03</b>	1.04E+02	1.07E+04
f13	1300	7.68E+03	1.31E+03	<b>1.31E+03</b>	4.03E+00	1.88E+01
f14	1400	4.18E+03	1.40E+03	<b>1.40E+03</b>	1.69E+00	7.69E+00
f15	1500	2.37E+04	1.50E+03	<b>1.50E+03</b>	7.24E-01	5.73E+01
f16	1600	1.59E+02	1.60E+03	<b>1.60E+03</b>	2.79E-01	-3.49E+00

f17	1700	8.40E+01	1.70E+03	<b>1.70E+03</b>	2.68E+00	-3.68E+00
f18	1800	1.79E+04	1.81E+03	<b>1.81E+03</b>	2.56E+00	3.44E+01
f19	1900	3.83E+04	1.90E+03	<b>1.90E+03</b>	4.15E-01	7.42E+01
f20	2000	1.08E+02	2.00E+03	<b>2.00E+03</b>	1.39E-01	-3.66E+00
f21	2100	4.78E+01	2.23E+03	<b>2.23E+03</b>	5.08E+01	-3.79E+00
f22	2200	5.88E+02	2.29E+03	<b>2.29E+03</b>	2.17E+01	-2.88E+00
f23	2300	2.87E+01	2.61E+03	<b>2.61E+03</b>	2.80E+00	-3.83E+00
f24	2400	1.47E+02	2.61E+03	<b>2.61E+03</b>	1.22E+02	-3.65E+00
f25	2500	5.02E+01	2.90E+03	<b>2.90E+03</b>	1.08E+01	-3.81E+00
f26	2600	3.42E+02	2.90E+03	<b>2.90E+03</b>	2.15E-10	-3.42E+00
f27	2700	3.94E+01	3.09E+03	<b>3.09E+03</b>	1.97E+00	-3.82E+00
f28	2800	1.08E+02	3.09E+03	<b>3.09E+03</b>	4.20E+01	-3.74E+00
f29	2900	9.40E+01	3.16E+03	<b>3.16E+03</b>	1.01E+01	-3.76E+00
f30	3000	3.75E+06	3.56E+03	<b>3.56E+03</b>	2.30E+02	4.08E+03

Table 11 describes the results obtained by applying the z test to the methods in comparison with DSFS with type-1 fuzzy logic vs HFPSO using 30 dimensions. Again the proposed method is significantly better.

Table 11. Results z-test for 30 dimensions.

HFPSO [40]				DSFS		
				Type 1 fuzzy logic		z
Function	fi	Mean	Std	Mean	Std	
f1	100	9.81E+08	1.01E+02	3.49E+03	2.76E+03	1.95E+06
f2	200	4.91E+08	2.00E+02	3.06E+16	7.11E+16	-2.36E+00
f3	300	5.96E+03	3.00E+02	8.40E+03	3.95E+03	-3.37E+00
f4	400	4.55E+01	4.01E+02	4.87E+02	3.56E+01	-6.01E+00
f5	500	1.84E+01	5.06E+02	6.11E+02	2.27E+01	-6.41E+00
f6	600	1.35E+01	6.00E+02	6.00E+02	1.36E-02	-5.35E+00
f7	700	1.73E+01	7.19E+02	8.53E+02	1.68E+01	-6.36E+00
f8	800	1.44E+01	8.07E+02	9.05E+02	2.12E+01	-6.04E+00
f9	900	3.07E+02	9.00E+02	9.01E+02	6.89E-01	-3.61E+00
f10	1000	3.79E+02	1.40E+03	6.21E+03	6.89E-01	-2.28E+01
f11	1100	5.24E+01	1.10E+03	1.19E+03	2.31E+01	-5.66E+00
f12	1200	4.13E+06	1.50E+03	1.56E+05	1.03E+05	2.11E+02
f13	1300	7.68E+03	1.31E+03	3.56E+03	9.09E+02	1.42E+01
f14	1400	4.18E+03	1.40E+03	1.50E+03	1.03E+01	1.05E+01
f15	1500	2.37E+04	1.50E+03	1.70E+03	4.43E+01	8.03E+01
f16	1600	1.59E+02	1.60E+03	2.45E+03	2.66E+02	-7.74E+00
f17	1700	8.40E+01	1.70E+03	1.85E+03	6.65E+01	-5.69E+00
f18	1800	1.79E+04	1.81E+03	2.87E+03	6.86E+02	4.25E+01

f19	1900	3.83E+04	1.90E+03	1.99E+03	1.88E+01	1.05E+02
f20	2000	1.08E+02	2.00E+03	2.39E+03	2.58E+01	-6.25E+00
f21	2100	4.78E+01	2.23E+03	2.40E+03	2.28E+01	-5.78E+00
f22	2200	5.88E+02	2.29E+03	2.30E+03	1.50E-02	-4.09E+00
f23	2300	2.87E+01	2.61E+03	2.74E+03	2.86E+01	-5.69E+00
f24	2400	1.47E+02	2.61E+03	2.91E+03	3.37E+01	-5.80E+00
f25	2500	5.02E+01	2.90E+03	2.89E+03	1.73E+00	-5.36E+00
f26	2600	3.42E+02	2.90E+03	4.28E+03	5.50E+02	-7.31E+00
f27	2700	3.94E+01	3.09E+03	3.22E+03	8.08E+00	-5.64E+00
f28	2800	1.08E+02	3.09E+03	3.21E+03	1.25E+01	-5.50E+00
f29	2900	9.40E+01	3.16E+03	3.63E+03	1.06E+02	-6.13E+00
f30	3000	3.75E+06	3.56E+03	1.32E+04	3.29E+03	4.22E+03

6. Conclusions

In conclusion we can describe stochastic methods, as metaheuristics that include basic algorithms for global stochastic optimization, such as random search, which helps the dispersion of the particles to reach the global optimum and avoid local stagnation. In this study, an improvement was made to a recent method that has been used in several optimization problems, this algorithm has two important processes for its operation, the diffusion and the updating process, which make up the most important part of Stochastic Fractal Search (SFS), this algorithm has been dynamically analyzed and improved with a fuzzy controller which controls the diffusion of particles and diversity by iteration, making this a better optimization method as has been observed in the results obtained compared to others. literature methods.

The experimentation was carried out first, in the comparison of the results of the Dynamic Stochastic Fractal Search (DFSF) method adjusted with both kinds of fuzzy logic, where we found that the values are very similar for all the CEC'2017 evaluation functions. After that, then a comparison with the combination the Hybrid firefly algorithm and particles swarm optimization (HFPSO) was also made showing a significant statistical advantage for the proposed DFSF method of this paper. In addition, since HFPSO was better than FA, PSO, HPSOFF and FFPSO [40], then the proposed DFSF is also better than these 5 metaheuristic optimization algorithms.

It was shown that the improvement applied to the method was satisfactory for its efficiency and optimization performance, this improvement was made by adding the iteration and diversity equations, to help the chaotic-stochastic movement of the particles, in addition, it was adjusted with a controller of fuzzy inference. Finally, it can be concluded that the combination of stochastic metaheuristics with fuzzy logic can generate good results for the efficiency and improvement of the optimization algorithms, as developed in this study. As future work, we plan to apply the proposed DFSF with real-world problems in different areas of application.

References

1. Holland, J. H. Genetic Algorithms understand Genetic Algorithms, *Sci. Am.*, 1992, vol. 267, no. 1, pp. 66–73.
2. Garg, Harish. A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data,. In Handbook of research on artificial intelligence techniques and algorithms. IGI Global, 2015. p 620-654.
3. Kennedy, J. and Eberhart, R. Particle swarm optimization,. In Neural Networks, The IEEE *International Conference on Neural Networks*. 1995, vol. 4, pp. 1942–1948.
4. Glover, F. and Laguna, M. *Tabu Search*. Springer US, 1997.
5. Gallego, R. A. Romero, A. and Monticelli, A. J. Tabu search algorithm for network synthesis, *IEEE Trans. Power Syst.*, 2000, vol. 15, no. 2, pp. 490–495.
6. Liu, B., Wang, L., Jin, Y. H. , Tang, F., and Huang, D. X. Improved particle swarm optimization combined with chaos,. *Chaos, Solitons & Fractals*, 2005, vol. 25, no 5, pp. 1261-1271.
7. Alatas, B. Akin, E. and Ozer, A. B. Chaos embedded particle swarm optimization algorithms,. *Chaos, Solitons and Fractals*, 2009, vol. 40, no. 4, pp. 1715–1734.
8. Jiao, B., Lian, Z., and Gu, X. A dynamic inertia weight particle swarm optimization algorithm,. *Chaos, Solitons and Fractals*, 2008, vol. 37, no. 3, pp. 698–705.
9. Yang, D., Li, G., and Cheng, G. On the efficiency of chaos optimization algorithms for global optimization,. *Chaos, Solitons & Fractals*, 2007, vol. 34, no 4, pp 1366-1375.
10. Misaghi, M., & Yaghoobi, M. Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller,. *Journal of computational Desing and Engineering*,. 2019, vol.6, no 3, pp 284-295.
11. Chen, B., Liu, X., and Tong, S,. Adaptive fuzzy approach to control unified chaotic systems,. *Chaos, Solitons and Fractals*, 2007, vol. 34, no. 4, pp. 1180–1187.
12. Lagunes, M. L., Castillo, O., Valdez, F., and Soria, J. *Comparative Study of Fuzzy Controller Optimization with Dynamic Parameter Adjustment Based on Type 1 and Type 2 Fuzzy Logic*,. In International Fuzzy Systems Association World Congress,. 2019, vol. 1000, pp 296-305.
13. Lagunes, M. L., Castillo, O., Valdez, F., Soria, J., and Melin, P,. Parameter Optimization for Membership Functions of Type-2 Fuzzy Controllers for Autonomous Mobile Robots Using the Firefly Algorithm,. In North american fuzzy information processing society annual conference. Springer,. 2018, pp. 569–579.
14. Amador-Angulo L. and Castillo, O. Comparative Analysis of Designing Differents Types of Membership Functions Using Bee Colony Optimization in the Stabilization of Fuzzy Controllers,. In Nature- inspired design of hybrid intelligent systems. Springer, 2017, pp. 551–571.
15. Lagunes, M. L., Castillo, O., Valdez, F., and Soria, J. *Comparison of Fuzzy Controller Optimization with Dynamic Parameter Adjustment Based on of Type-1 and Type-2 Fuzzy Logic*,. In Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine. 2020, vol. 827, pp. 47-56.
16. Pérez, J., Valdez, F., and Castillo, O. Modification of the Bat Algorithm Using Type-2 Fuzzy Logic for Dynamical Parameter Adaptation,. In Nature- inspired design of hybrid intelligent systems. Springer,. 2017, pp. 343–355.
17. Lagunes, M. L., Castillo, O., Valdez, F., Soria, J., and Melin, P. Parameter Optimization for Membership Functions of Type-2 Fuzzy Controllers for Autonomous Mobile Robots Using the Firefly Algorithm,. In North american fuzzy information processing society annual conference.

- Springer, 2018, pp. 569–579.
18. Bernal, E., Castillo, O., Soria, J., Valdez, F., and Melin, P., A variant to the dynamic adaptation of parameters in galactic swarm optimization using a fuzzy logic augmentation,” In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018, pp. 1–7.
  19. Sánchez, D., Melin, P., and Castillo, O., Optimization of modular granular neural networks using a firefly algorithm for human recognition,” *Eng. Appl. Artif. Intell.*, 2017, vol. 64, pp. 172–186.
  20. Mandelbrot, B., The fractal geometry of nature,. 1982, vol. 1.
  21. Barnsley, M. F., and Demko, S. Iterated function systems and the global construction of fractals., *Proc. R. Soc. London, Ser. A Math. Phys. Sci.*, 1985, vol. 399, no. 1817, pp. 243–275.
  22. Grassberger, P., and Procaccia, I. Characterization of strange attractors., *Phys. Rev. Lett.*, 1983. vol. 50, no. 5, pp. 346–349.
  23. Prusinkiewicz, P., Graphical applications of L-systems. In *Proceedings of graphics interface*. 1986, vol 86. no 86, pp 247-253.
  24. Rushton, B., Subdivision rules for all Gromov hyperbolic groups, 2017.
  25. Falconer, K. J. Random fractals,. *Math. Proc. Cambridge Philos. Soc.*, 1986, vol. 100, no. 3, pp. 559–582.
  26. Khalilpourazari S., and S. Khalilpourazary, S., A Robust Stochastic Fractal Search approach for optimization of the surface grinding process,. *Swarm Evol. Comput.*, 2018, vol. 38, pp. 173–186.
  27. Mellal, M. A. and Zio, E., A penalty guided stochastic fractal search approach for system reliability optimization,. *Reliab. Eng. Syst. Saf.*, 2016, vol. 152, pp. 213–227.
  28. Salimi, H., Stochastic Fractal Search: A powerful metaheuristic algorithm, *Knowledge-Based Syst.*, 2015, vol. 75, pp. 1–18.
  29. S. Khalilpourazari, S., Naderi, B., and Khalilpourazary, S., Multi-Objective Stochastic Fractal Search: a powerful algorithm for solving complex multi-objective optimization problems,. *Soft Comput.*, 2020, vol. 24, no. 4, pp. 3037–3066.
  30. Aras, S., Gedikli, E., and Kahraman, H. T., A novel stochastic fractal search algorithm with fitness-Distance balance for global numerical optimization,” *Swarm Evol. Comput.*, 2021. vol. 61, p. 100821.
  31. Çelik, E., Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system,” *Neural Comput. Appl.*, 2018, vol. 30, no. 6, pp. 1991–2002.
  32. Alomoush, M. I. and Oweis, Z. B., Environmental-economic dispatch using stochastic fractal search algorithm,” *Int. Trans. Electr. Energy Syst.*, 2018, vol. 28, no. 5, p. e2530.
  33. Miramontes, I., Melin, P., and Prado-Arechiga, G. Comparative Study of Bio-inspired Algorithms Applied in the Optimization of Fuzzy Systems, In *Hybrid intelligent Systems in Control, Pattern Recognition and Medicine*, Springer., 2020, vol. 827, pp. 219–231.
  34. Miramontes, I., Melin, P., and Prado-Arechiga, G. Fuzzy System for Classification of Nocturnal Blood Pressure Profile and Its Optimization with the Crow Search Algorithm, in *Advances in Intelligent Systems and Computing*, 2021, vol. 1222 AISC, pp. 23–34.
  35. Guzmán, J., Miramontes, I., Melin, P., and Prado-Arechiga, G. Optimal Genetic Design of Type-1 and Interval Type-2 Fuzzy Systems for Blood Pressure Level Classification,” *Axioms*, 2019, vol. 8, no. 1, pp. 8.
  36. Carvajal, O., Melin, P., Miramontes, I, and Prado-Arechiga, G. Optimal design of a general type-2

- 
- fuzzy classifier for the pulse level and its hardware implementation,. *Eng. Appl. Artif. Intell.*, 2021, vol. 97, pp. 104069.
37. Miramontes, I., Guzman, J.C., Melin, P., and Prado-Arechiga, G. Optimal design of interval type-2 fuzzy heart rate level classification systems using the bird swarm algorithm. *Algorithms*, 2018, vol. 11, no. 12, p. 206.
38. Biedrzycki, R., A Version of IPOP-CMA-ES Algorithm with Midpoint for CEC 2017 Single Objective Bound Constrained Problems, In 2017 Congress on Evolutionary Computatio,. 2017, pp 1489-1494.
39. Awad, N. H, et al. Evaluation Criteria for the CEC 2017 Special session and competition on single obejective real- parameter numerical optimization *Nanyang Technol. Univ., Singapore, Jordan Univ. Sci. Technol., Jordan Zhengzhou Univ., Zhengzhou, China, Tech. Rep*, 2016.
40. Aydilek, İ. B. A Hybrid Firefly and Particle Swarm Optimization Algorithm for Computationally Expensive Numerical Problems., *Appl. Soft Comput.*, 2018, vol. 66, pp. 232–249.