



Marylu L. Lagunes, Oscar Castillo * D, Fevrier Valdez D, Jose Soria and Patricia Melin D

Tijuana Institute of Technology, Calzada Tecnologico s/n, Fracc. Tomas Aquino, Tijuana 22379, Mexico; marylu.lara@tectijuana.edu.mx (M.L.L.); fevrier@tectijuana.mx (F.V.); jsoria57@gmail.com (J.S.); pmelin@tectijuana.mx (P.M.)

* Correspondence: ocastillo@tectijuana.mx

Abstract: 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 of searching for the local optimum and are not very effective for finding optimal solutions in real problems. The multi-metaheuristic models are those formed by more than two metaheuristics to solve a common optimization problem where each of the metaheuristics is used as an optimization tool. The chosen metaheuristic depends on the problem to be solved; most of the metaheuristic methods have the same inspiration and goal means, and generally, their performance is better than a simple heuristic. All metaheuristic algorithms use certain trade-offs between randomization and local search. These algorithms have two main characteristics, intensification and diversification. Diversification refers to generating solutions to explore the search space on a global scale, and intensification refers to focusing on the search for a local region until the best region is found. The combination of these two features results in global optimization [1]. 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) [2,3], which was inspired by biological evolution, with random genetic combinations and mutations in a chromosome, and is based on the selection, crossover, mutation, and replacement operators. The inspiration for particle swarm optimization (PSO) was natural fish and bird behavior when moving in swarms; each particle moves randomly to find the global optimum, updating its speed and position until finding the best global solution [4]. Tabu search (TS) [5,6] 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 [7], an algorithm for optimization of hybrid particle swarms incorporating chaos is proposed, where the adaptive inertia weight factor (AIWF) is used for enhancing the PSO in balancing, in an efficient way, the diversification and intensification abilities of the algorithm. In this case, the hybridization of PSO with AIWF



Citation: Lagunes, M.L.; Castillo, O.; Valdez, F.; Soria, J.; Melin, P. A New Approach for Dynamic Stochastic Fractal Search with Fuzzy Logic for Parameter Adaptation. *Fractal Fract.* 2021, *5*, 33. https://doi.org/10.3390/ fractalfract5020033

Academic Editors: Ángel Alberto Magreñán and Iñigo Sarría

Received: 25 March 2021 Accepted: 15 April 2021 Published: 17 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). and chaos was used to build a chaotic PSO (CPSO) by prudently combining the evolutionary search ability based on the population of the PSO and the behavior of the chaotic search. Furthermore, [8] proposes a new PSO algorithm that relies on chaotic equation maps for parameter adaptation, and this is done through the use of chaotic number generators every time the classical PSO algorithm needs a random number [9]. On the other hand, [10] developed an improved particle swarm optimization (IPSO) algorithm for enhancing the performance of the traditional PSO, which uses a dynamic inertia weight. In [11], the authors present an invasive weed optimization (IWO) metaheuristic, which is a weed colony-based population optimization approach also based on chaos theory. In addition to the improvements to stochastic optimization methods, there are also combinations or hybridizations with fuzzy logic to improve performance or reach a specific solution. For example, in [12], an adaptive fuzzy control approach to control chaotic unified systems was proposed. In addition, in [13–16], a multi-metaheuristic model was developed for the optimal design of fuzzy controllers. An enhancement to the Bat algorithm was carried out in [17] for the dynamic adaptation of parameters. There are different applications using metaheuristics and fuzzy logic for optimization [18,19]. 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 was considered by analyzing its chaotic stochastic characteristics in the diffusion process, where each of its particles was generated and moved in a random stochastic way. Second, it was detected that the stochastic movement of each particle may not be optimal due to the formation of the fractal not being able to achieve exploration and exploitation of the entire search space. Therefore, diversity was introduced for each iteration to eventually get closer to the global optima by looking for the particles with the best fitness and 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 of this article is 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 of the search space. Thus, it was decided to add dynamic adjustment to the mentioned parameter to achieve better movement in each of the newly generated fractal particles; this improvement was implemented using type-1 and type-2 fuzzy systems. This was done by making dynamic changes with a controller, which had diversity as Input 1 and iteration as Input 2. Diversification is a charge of spreading the particles throughout the search area for each iteration. As a result, a dynamically adapted method was obtained, which does not stagnate as fast in the optimal local ones, thus reaching the global optimum, and therefore, improving the effectiveness of the dynamic stochastic fractal search (DSFS) method. The motivation for the development of this article was the creation of a stochastic fractal method with dynamic adjustment that does not remain stagnant in the local optimum and achieves reaching the global optimum of the objective function. To comply with this, it was designed with a fuzzy inference system that controls the diffusion of the particles in each iteration through diversity. Applied in this way, the method has less chance of stagnation and premature convergence. The goal was to build an efficient but practical algorithm that works most of the time and is capable of producing good quality solutions for finding the overall optimal solution of real-world problems.

The rest of this article is structured as follows. 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 for Stochastic Fractal Search (SFS)

The term "fractal" was used for the first time by Benoit Mandelbrot [20], who described it, in his theory of fractals, as 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 can 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 is 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. 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, it avoids 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. Equation (1) establishes how to generate an initial population for a particular problem based on its constraints:

$$P = LB + \varepsilon * (UP - LB) \tag{1}$$

where the particle population *P* is randomly produced, considering the problem constraints after setting the lower (*LB*) and the upper (*UB*) limits, and ε represents a number randomly produced in the range 0–1.

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

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

$$GW_2 = \text{Gaussian}\left(\mu_P, \sigma\right) \tag{3}$$

where ε and ε' represent randomly generated numbers in the range of 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 μ and a standard deviation σ :

$$\sigma = \frac{\log g}{g} \times |P_i - BP| \tag{4}$$

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

The update process (representing exploration in a fractal search) is expressed as follows:

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

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*.

$$P'_{i}(j) = P_{x}(j) - \varepsilon \times \left(P_{y}(j) - P_{i}(j)\right)$$
(6)

where the augmented component is given by $P'_i(j)$, and P_x and 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.

$$P'_{i} = P_{i} - \varepsilon \times (P_{x} - BP) | \varepsilon' \le 0.5$$
(7)

$$P'_{i} = P_{i} - \varepsilon \times (P_{x} - P_{y}) | \text{otherwise}$$
(8)

Once the first updating stage is finished, the second one is initiated with a ranking of all the points based on Equations (7) and (8). As previously mentioned, if Pa_i is lower than a random ε , 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 a 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. In analyzing the diffusion method, we know that a seed particle is initialized, and others are generated and adhere to it through diffusion, which forms a chaotic stochastic fractal branch. By taking this fact into account, the stochastic fractal search (SFS) method has been improved by adding diversity to the diffusion process for the particles in each iteration. In this way, the Gaussian random walk was improved because the particles have more possibilities to exploit the search space and therefore do not stagnate in a suboptimal location. To control the diffusion parameter, a fuzzy inference system was introduced, which dynamically adjusted the diffusion of the particles in a range of 0–1. This fuzzy system for control has two inputs (iteration and diversity) and one output (diffusion), as illustrated in Figures 1 and 2. The implemented fuzzy system has 9 if-then rules, which allow for the expression of the knowledge available about the relationship between the antecedents and consequents. These rules determine the behavior of the fuzzy controller and it is here that the output parameter is emulated. The idea was to have an efficient algorithm that improves the disadvantages of the SFS, which is capable of producing solutions of good quality. In this case, Equations (11) and (12) represent diversity and iteration, respectively.



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





The fuzzy if-then rules are as follows:

- 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).

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

$$triangular(u;a,b,c) = \begin{cases} 0, & u \le a \\ \frac{u-a}{b-c}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}$$
(9)

$$\begin{split} \widetilde{\mu}(x) &= \left[\underline{\mu}(x), \overline{\mu}(x)\right] = \text{itrapatype2}(x, [a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, \alpha]) \\ \text{where } a_1 < a_2, \ b_1 < b_2, \ c_1 < c_2, \ d_1 < d_2. \\ \mu_1(x) &= \max\left(\min\left(\frac{x-a_1}{b_1-a_1}, 1, \frac{d_1-x}{d_1-c_1}\right), 0\right) \\ \mu_2(x) &= \max\left(\min\left(\frac{x-a_2}{b_2-a_2}, 1, \frac{d_2-x}{d_2-c_2}\right), 0\right) \\ \overline{\mu}(x) &= \begin{cases} \max(\mu_1(x), \mu_2(x)) & \forall \ x \notin (b1, c2) \\ 1 & \forall \ x \in (b1, c2) \\ 1 & \forall \ x \in (b1, c2) \end{cases} \\ \underline{\mu}(x) &= \min(\alpha, \min(\mu_1(x), \mu_2(x))) \\ \text{Iteration} &= \frac{Current \ Iteration}{Total \ of \ Iterations} \end{split}$$
(11)

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

Diversity contributes to the stochastic movement of the particles by having more possibilities of exploiting the entire search space. In this case, in each iteration of the search process, the parameter is dynamically adjusted close to the global optimum, and thus, the diffusion process is improved and we have a more efficient search method.

4. Experimental Results

To obtain an understanding of the effectiveness of the proposed dynamic stochastic fractal search (DSFS) method, 30 optimization functions of the Congress on Evolutionary Computing 2017 (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 with respect to other methods, different numbers of dimensions (10, 30, 50, and 100) and different adaptation strategies for the parameters (using type-1 and type-2 fuzzy systems) were used.

Table 1. CEC 2017 optimization functions.

| Type of Function | No | Name of Function | | | |
|---------------------------------------|----|---|------|--|--|
| · · · · · · · · · · · · · · · · · · · | 1 | Shifted and Rotated Bent Cigar Function | 100 | | |
| Functions | 2 | Shifted and Rotated Sum of Different Power Function | 200 | | |
| | 3 | Shifted and Rotated Zakharov Function | 300 | | |
| | 4 | Shifted and Rotated Rosenbrock's Function | 400 | | |
| | 5 | Shifted and Rotated Rastrigin's Function | 500 | | |
| Simple | 6 | Shifted and Rotated Expanded Schaffer's Function | 600 | | |
| Multimodal | 7 | Shifted and Rotated Lunacek Bi-Rastrigin Function | 700 | | |
| Functions | 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 | | |
| | 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 | | |
| ** 1 . 1 | 15 | Hybrid Function 5 ($N = 4$) | 1500 | | |
| Hybrid | 16 | Hybrid Function 6 ($N = 4$) | 1600 | | |
| 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 | | |
| Composition | 26 | Composition Function 6 ($N = 3$) | 2600 | | |
| Functions | 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] were considered in the tests. Table 2 shows, at the top, the number of dimensions used for each function. Column 1 shows the function that is given as f1, f2, f3, f4, ..., f30, Column 2, represented as "fi" shows the global optimum (minimum) of each function. The next two columns of Table 2 contain the obtained results 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.

| Dynamic Stochastic Fractal Search (DSFS) with 10 Dimensions | | | | | | | | |
|---|------|--------------------|-----------------------|--------------------|-----------------------|--|--|--|
| | | Type-1 F | uzzy Logic | Type-2 F | uzzy Logic | | | |
| Function | fi | Main | Std | Main | Std | | | |
| f1 | 100 | 1.01×10^2 | $4.87	imes10^{-1}$ | 1.01×10^2 | $7.85	imes10^{-1}$ | | | |
| f2 | 200 | $2.00 	imes 10^2$ | 0.00 | $2.00 	imes 10^2$ | 0.00 | | | |
| f3 | 300 | $3.00 	imes 10^2$ | $3.73	imes10^{-6}$ | $3.00 	imes 10^2$ | $5.04	imes10^{-6}$ | | | |
| f4 | 400 | $4.01 	imes 10^2$ | $7.81	imes10^{-1}$ | $4.00 	imes 10^2$ | $6.19	imes10^{-1}$ | | | |
| f5 | 500 | $5.06 	imes 10^2$ | 2.04 | $5.07 	imes 10^2$ | 2.64 | | | |
| f6 | 600 | $6.00 	imes 10^2$ | $7.21 	imes 10^{-8}$ | 6.00×10^2 | $6.24	imes10^{-8}$ | | | |
| f7 | 700 | $7.19	imes10^2$ | 3.45 | 7.20×10^2 | 3.32 | | | |
| f8 | 800 | $8.07	imes10^2$ | 3.04 | $8.07 	imes 10^2$ | 2.44 | | | |
| f9 | 900 | $9.00 	imes 10^2$ | 0.00 | $9.00 	imes 10^2$ | 0.00 | | | |
| f10 | 1000 | $1.40 	imes 10^3$ | 1.70×10^2 | $1.34	imes10^3$ | $1.31 	imes 10^2$ | | | |
| f11 | 1100 | $1.10 	imes 10^3$ | $9.31	imes10^{-1}$ | $1.10 	imes 10^3$ | 1.00 | | | |
| f12 | 1200 | $1.50 	imes 10^3$ | $1.04	imes10^2$ | $1.52 	imes 10^3$ | $1.02 	imes 10^2$ | | | |
| f13 | 1300 | $1.31 	imes 10^3$ | 4.03 | $1.31 	imes 10^3$ | 4.14 | | | |
| f14 | 1400 | $1.40 	imes 10^3$ | 1.69 | $1.40 	imes 10^3$ | 2.01 | | | |
| f15 | 1500 | $1.50 	imes 10^3$ | $7.24	imes10^{-1}$ | $1.50 	imes 10^3$ | $8.88	imes10^{-1}$ | | | |
| f16 | 1600 | $1.60 	imes 10^3$ | $2.79	imes10^{-1}$ | $1.60 	imes 10^3$ | $3.64	imes10^{-1}$ | | | |
| f17 | 1700 | $1.70 	imes 10^3$ | 2.68 | $1.71 	imes 10^3$ | 4.41 | | | |
| f18 | 1800 | $1.81 	imes 10^3$ | 2.56 | $1.81 	imes 10^3$ | 2.57 | | | |
| f19 | 1900 | $1.90 	imes 10^3$ | $4.15	imes10^{-1}$ | $1.90 	imes 10^3$ | $4.45	imes10^{-1}$ | | | |
| f20 | 2000 | $2.00 	imes 10^3$ | $1.39	imes10^{-1}$ | $2.00 	imes 10^3$ | $4.73	imes10^{-3}$ | | | |
| f21 | 2100 | 2.23×10^3 | 5.08 	imes 10 | $2.24 	imes 10^3$ | 5.40 	imes 10 | | | |
| f22 | 2200 | 2.29×10^3 | 2.17×10 | $2.28 	imes 10^3$ | 3.93×10 | | | |
| f23 | 2300 | 2.61×10^3 | 2.80 | $2.60 	imes 10^3$ | 5.61×10 | | | |
| f24 | 2400 | $2.61 	imes 10^3$ | 1.22×10^2 | $2.63 	imes 10^3$ | 1.14×10^2 | | | |
| f25 | 2500 | $2.90 	imes 10^3$ | 1.08 	imes 10 | $2.90 	imes 10^3$ | 1.15 	imes 10 | | | |
| f26 | 2600 | 2.90×10^3 | $2.15 	imes 10^{-10}$ | $2.90 	imes 10^3$ | $3.02 	imes 10^{-10}$ | | | |
| f27 | 2700 | 3.09×10^3 | 1.97 | $3.09 	imes 10^3$ | 2.14 | | | |
| f28 | 2800 | 3.09×10^3 | 4.20×10 | $3.11 	imes 10^3$ | 5.69 	imes 10 | | | |
| f29 | 2900 | $3.16 	imes 10^3$ | 1.01×10 | $3.16 	imes 10^3$ | 9.65 | | | |
| f30 | 3000 | 3.56×10^3 | $2.30 	imes 10^2$ | $3.55 	imes 10^3$ | $1.19	imes10^2$ | | | |

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

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. In Table 3, Row 3, the first column shows the number of the function that is being evaluated. In Column 2, the optimum of each function is observed. In Column 3, the average results of each of the functions are given, and Column 4 shows the standard deviation; these two were

obtained with type-1 fuzzy logic. Finally, the last columns illustrate the results in average and standard deviation, respectively, with type-2 fuzzy logic. It can be appreciated that the difference between these results was minimal, even though these results were obtained with experimentation using 30 dimensions. With this number of dimensions, the algorithm has a wider search space. In addition, the functions that were optimized are complex. Despite this situation, the method showed that it was approaching the optimum of each function, and the values obtained with both types of fuzzy logic were also very close.

| Dynamic Stochastic Fractal Search (DFSF) with 30 Dimensions | | | | | | | | |
|---|------|----------------------|-----------------------|----------------------|----------------------|--|--|--|
| | | Type-1 F | uzzy Logic | Type-2 Fi | uzzy Logic | | | |
| Function | fi | Main | Std | Main | Std | | | |
| f1 | 100 | $3.49 	imes 10^3$ | 2.76×10^3 | $3.24 	imes 10^3$ | $2.74 	imes 10^3$ | | | |
| f2 | 200 | $3.06	imes10^{16}$ | $7.11 	imes 10^{16}$ | $7.59	imes10^{17}$ | $3.98	imes10^{18}$ | | | |
| f3 | 300 | $8.40	imes10^3$ | $3.95 	imes 10^3$ | $8.79	imes10^3$ | $4.60 	imes 10^3$ | | | |
| f4 | 400 | $4.87 	imes 10^2$ | 3.56×10 | $4.80 	imes 10^2$ | 3.39×10 | | | |
| f5 | 500 | $6.11 	imes 10^2$ | 2.27 	imes 10 | $6.00 	imes 10^2$ | 2.02×10 | | | |
| f6 | 600 | $6.00 	imes 10^2$ | $1.36	imes10^{-2}$ | $6.00 	imes 10^2$ | $1.13	imes10^{-2}$ | | | |
| f7 | 700 | $8.53 	imes 10^2$ | 1.68 	imes 10 | $8.60 	imes 10^2$ | 1.70 	imes 10 | | | |
| f8 | 800 | $9.05 	imes 10^2$ | 2.12×10 | $9.06 	imes 10^2$ | 2.27 	imes 10 | | | |
| f9 | 900 | $9.01 	imes 10^2$ | $6.89	imes10^{-1}$ | $9.04	imes10^2$ | 1.44	imes10 | | | |
| f10 | 1000 | $6.21 	imes 10^3$ | $6.89	imes10^{-1}$ | $6.05 	imes 10^3$ | $5.03 	imes 10^2$ | | | |
| f11 | 1100 | $1.19	imes10^3$ | 2.31×10 | $1.19	imes10^3$ | 2.90×10 | | | |
| f12 | 1200 | $1.56 	imes 10^5$ | $1.03	imes10^5$ | $1.81 	imes 10^5$ | $1.48 	imes 10^5$ | | | |
| f13 | 1300 | $3.56 	imes 10^3$ | $9.09 	imes 10^2$ | $4.01 	imes 10^3$ | $1.00 	imes 10^3$ | | | |
| f14 | 1400 | $1.50 	imes 10^3$ | 1.03×10 | $1.50 	imes 10^3$ | 1.06 	imes 10 | | | |
| f15 | 1500 | $1.70 	imes 10^3$ | 4.43×10 | $1.70 	imes 10^3$ | 3.95 	imes 10 | | | |
| f16 | 1600 | $2.45 	imes 10^3$ | $2.66 	imes 10^2$ | $2.47 	imes 10^3$ | $2.14 	imes 10^2$ | | | |
| f17 | 1700 | $1.85 	imes 10^3$ | 6.65×10 | $1.86 	imes 10^3$ | 8.06×10 | | | |
| f18 | 1800 | $2.87 	imes 10^3$ | $6.86 	imes 10^2$ | $2.69 	imes 10^3$ | 3.69×10^{2} | | | |
| f19 | 1900 | $1.99 	imes 10^3$ | 1.88 	imes 10 | $1.99 	imes 10^3$ | 1.87 	imes 10 | | | |
| f20 | 2000 | 2.39×10^{3} | 2.58×10 | 2.22×10^3 | $1.12 	imes 10^2$ | | | |
| f21 | 2100 | 2.40×10^{3} | 2.28×10 | 2.39×10^{3} | 2.51×10 | | | |
| f22 | 2200 | 2.30×10^{3} | 1.50×10^{-2} | 2.30×10^{3} | $7.44	imes10^{-3}$ | | | |
| f23 | 2300 | $2.74 	imes 10^3$ | 2.86×10 | 2.73×10^{3} | 2.26×10 | | | |
| f24 | 2400 | 2.91×10^{3} | 3.37×10 | 2.90×10^{3} | 3.04×10 | | | |
| f25 | 2500 | 2.89×10^{3} | 1.73 | 2.89×10^{3} | $9.76	imes10^{-1}$ | | | |
| f26 | 2600 | $4.28 	imes 10^3$ | 5.50×10^2 | 4.22×10^3 | 6.59×10^{2} | | | |
| f27 | 2700 | 3.22×10^{3} | 8.08 | 3.22×10^{3} | 7.37 | | | |
| f28 | 2800 | 3.21×10^{3} | 1.25×10 | 3.21×10^3 | 1.27×10 | | | |
| f29 | 2900 | 3.63×10^3 | 1.06×10^2 | 3.59×10^3 | $1.13 	imes 10^2$ | | | |
| f30 | 3000 | $1.32 	imes 10^4$ | 3.29×10^3 | $1.51 	imes 10^4$ | $5.55 	imes 10^3$ | | | |

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

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 at the global optimal solution. In this study, not only unimodal and multimodal functions were optimized but also hybrid and complex functions were 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 the adaptation of parameters show that the method had some degree of difficulty in reaching the global optimum. Even so, they provided good approximation values, showing that the improved method is efficient in optimization tasks. The explanation of the rows and columns of Tables 4 and 5 is analogous to the previously mentioned description for Table 3.

| Dynamic Stochastic Fractal Search (DSFS) with 50 Dimensions | | | | | | | |
|---|------|----------------------|----------------------|----------------------|----------------------|--|--|
| | | Type-1 Fu | Type-1 Fuzzy Logic | | ızzy Logic | | |
| Function | fi | Main | Std | Main | Std | | |
| f1 | 100 | $9.00 	imes 10^4$ | $4.30 	imes 10^4$ | $8.42 	imes 10^4$ | $6.69 	imes 10^4$ | | |
| f2 | 200 | $1.20	imes10^{40}$ | $4.53	imes10^{40}$ | $5.08 	imes 10^{39}$ | $1.52 	imes 10^{40}$ | | |
| f3 | 300 | $6.01	imes10^4$ | $9.72 	imes 10^3$ | $6.28	imes10^4$ | $1.39	imes10^4$ | | |
| f4 | 400 | $5.72 	imes 10^2$ | 4.05 	imes 10 | $5.69 	imes 10^2$ | 4.26×10 | | |
| f5 | 500 | $7.68 	imes 10^2$ | 4.35×10 | 7.69×10^2 | 4.59 	imes 10 | | |
| f6 | 600 | $6.01 	imes 10^2$ | $1.35	imes10^{-1}$ | 6.01×10^2 | $1.55	imes10^{-1}$ | | |
| f7 | 700 | $1.05 	imes 10^3$ | 3.18 	imes 10 | $1.06 	imes 10^3$ | 2.61×10 | | |
| f8 | 800 | $1.06 	imes 10^3$ | 3.69×10 | $1.06 	imes 10^3$ | 4.42×10 | | |
| f9 | 900 | $1.13 	imes 10^3$ | $1.49 	imes 10^2$ | $1.14	imes10^3$ | 1.22×10^2 | | |
| f10 | 1000 | $1.13	imes10^4$ | $4.84	imes10^2$ | $1.12	imes10^4$ | $6.53	imes10^2$ | | |
| f11 | 1100 | $1.36 	imes 10^3$ | 3.00 	imes 10 | $1.36 	imes 10^3$ | 4.30 	imes 10 | | |
| f12 | 1200 | $3.54	imes10^6$ | $1.42 	imes 10^6$ | $3.47	imes10^6$ | $1.96	imes10^6$ | | |
| f13 | 1300 | $1.48	imes10^4$ | $1.17	imes10^4$ | $1.67	imes10^4$ | $1.36	imes10^4$ | | |
| f14 | 1400 | $1.50 	imes 10^3$ | 1.03 	imes 10 | $1.80 	imes 10^3$ | 8.36 	imes 10 | | |
| f15 | 1500 | $4.14 	imes 10^3$ | $1.54 	imes 10^3$ | $3.75 	imes 10^3$ | $8.43 	imes 10^2$ | | |
| f16 | 1600 | 3.52×10^3 | $3.87 	imes 10^2$ | $3.58 	imes 10^3$ | $4.80 	imes 10^2$ | | |
| f17 | 1700 | $3.05 	imes 10^3$ | 2.37×10^{2} | $3.04 	imes 10^3$ | $2.56 	imes 10^2$ | | |
| f18 | 1800 | $5.42 	imes 10^4$ | $3.34	imes10^4$ | $5.39 	imes 10^4$ | $3.20 	imes 10^4$ | | |
| f19 | 1900 | $7.10 	imes 10^3$ | $4.05	imes10^3$ | $7.73 	imes 10^3$ | $5.45 	imes 10^3$ | | |
| f20 | 2000 | $3.10 	imes 10^3$ | 2.16×10^{2} | $3.14 	imes 10^3$ | 2.36×10^{2} | | |
| f21 | 2100 | $2.55 	imes 10^3$ | 4.55 	imes 10 | $2.55 	imes 10^3$ | 4.99 	imes 10 | | |
| f22 | 2200 | $1.10	imes10^4$ | $4.12 	imes 10^3$ | $1.23	imes10^4$ | $3.40 	imes 10^3$ | | |
| f23 | 2300 | $3.00 	imes 10^3$ | 4.43×10 | $2.98 	imes 10^3$ | 4.91×10 | | |
| f24 | 2400 | $3.14 	imes 10^3$ | 5.98×10 | $3.14 	imes 10^3$ | 5.85 	imes 10 | | |
| f25 | 2500 | $3.07 	imes 10^3$ | 2.52×10 | $3.08 	imes 10^3$ | 2.22×10 | | |
| f26 | 2600 | $6.07 	imes 10^3$ | 5.92×10^{2} | $6.16 	imes 10^3$ | $4.77 	imes 10^2$ | | |
| f27 | 2700 | 3.41×10^3 | 4.40×10 | $3.40 	imes 10^3$ | 4.09×10 | | |
| f28 | 2800 | $3.36 	imes 10^3$ | 3.36×10 | $3.35 	imes 10^3$ | 3.44×10 | | |
| f29 | 2900 | $4.15 	imes 10^3$ | 2.71×10^2 | $4.17 	imes 10^3$ | 2.55×10^2 | | |
| f30 | 3000 | 3.25×10^{6} | 6.84×10^{5} | 3.24×10^{6} | 8.20×10^{5} | | |

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

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

| Dynamic Stochastic Fractal Search DSFS with 100 Dimensions | | | | | | | |
|--|------|----------------------|---------------------|----------------------|---------------------|--|--|
| | | Type-1 Fu | uzzy Logic | Type-2 Fi | uzzy Logic | | |
| Function | fi | Main | Std | Main | Std | | |
| f1 | 100 | $1.15 	imes 10^8$ | $4.36 	imes 10^7$ | $1.08 	imes 10^8$ | $3.31 	imes 10^7$ | | |
| f2 | 200 | $1.34	imes10^{108}$ | $8.85	imes10^{108}$ | $1.14	imes 10^{111}$ | $5.81	imes10^{111}$ | | |
| f3 | 300 | $2.53 	imes 10^5$ | $2.78	imes10^4$ | $2.56	imes10^5$ | $2.78	imes10^4$ | | |
| f4 | 400 | 9.23×10^{2} | 4.74	imes10 | $9.37 	imes 10^2$ | 6.10 	imes 10 | | |
| f5 | 500 | $1.29 	imes 10^3$ | 9.15 	imes 10 | $1.29 	imes 10^3$ | 7.11×10 | | |
| f6 | 600 | $6.07 	imes 10^2$ | 1.33 | $6.07	imes10^2$ | 1.02 | | |
| f7 | 700 | $1.72 	imes 10^3$ | 5.61 	imes 10 | $1.73 	imes 10^3$ | 5.30×10 | | |
| f8 | 800 | $1.58 	imes 10^3$ | 7.70 	imes 10 | $1.59 	imes 10^3$ | 8.69 	imes 10 | | |
| f9 | 900 | $1.19	imes 10^4$ | $3.07	imes10^3$ | $1.28	imes10^4$ | $4.21 	imes 10^3$ | | |
| f10 | 1000 | $2.71 	imes 10^4$ | $1.04	imes10^3$ | $2.72 	imes 10^4$ | $9.91 	imes 10^2$ | | |
| f11 | 1100 | $1.62 	imes 10^4$ | $3.86 	imes 10^3$ | $1.63	imes10^4$ | $4.35 	imes 10^3$ | | |
| f12 | 1200 | $6.66 	imes 10^7$ | $2.09	imes10^7$ | $7.05	imes10^7$ | $1.74	imes10^7$ | | |
| f13 | 1300 | $5.66 	imes 10^3$ | $1.80 	imes 10^3$ | $6.16 	imes 10^3$ | 2.89×10^3 | | |
| f14 | 1400 | $3.36 	imes 10^5$ | $2.50 	imes 10^5$ | $2.36 	imes 10^5$ | $1.44 	imes 10^5$ | | |
| f15 | 1500 | 4.46×10^3 | 4.66×10^3 | $4.39 	imes 10^3$ | $3.13 	imes 10^3$ | | |

| Dynamic Stochastic Fractal Search DSFS with 100 Dimensions | | | | | | | | |
|--|------|--------------------|-------------------|-------------------|--------------------|--|--|--|
| f16 | 1600 | $7.97 	imes 10^3$ | $8.58 	imes 10^2$ | $7.70 	imes 10^3$ | $7.93 	imes 10^2$ | | | |
| f17 | 1700 | $6.04 	imes 10^3$ | $3.39	imes10^2$ | $5.90 	imes 10^3$ | $5.76 	imes 10^2$ | | | |
| f18 | 1800 | $5.97 	imes 10^5$ | $3.86 	imes 10^5$ | $6.45 	imes 10^5$ | $3.99	imes10^5$ | | | |
| f19 | 1900 | 3.60×10^3 | $1.65 	imes 10^3$ | $3.49 	imes 10^3$ | $1.56 	imes 10^3$ | | | |
| f20 | 2000 | $2.18 	imes 10^3$ | 8.28 	imes 10 | $6.27 	imes 10^3$ | $4.30 	imes 10^2$ | | | |
| f21 | 2100 | $3.11 	imes 10^3$ | 6.75 	imes 10 | $3.11 	imes 10^3$ | 6.06×10 | | | |
| f22 | 2200 | $2.96	imes10^4$ | $8.67 	imes 10^2$ | $2.97	imes10^4$ | 8.23×10^2 | | | |
| f23 | 2300 | $3.59 	imes 10^3$ | 5.69 	imes 10 | $3.57 	imes 10^3$ | 7.87 	imes 10 | | | |
| f24 | 2400 | $4.09 	imes 10^3$ | 9.76	imes10 | $4.08 	imes 10^3$ | $1.19 	imes 10^2$ | | | |
| f25 | 2500 | $3.63	imes10^3$ | 6.17	imes10 | $3.63	imes10^3$ | 5.46 	imes 10 | | | |
| f26 | 2600 | $1.41 	imes 10^4$ | $1.04	imes10^3$ | $1.42 	imes 10^4$ | $8.28 	imes 10^2$ | | | |
| f27 | 2700 | 3.72×10^3 | 7.06 	imes 10 | $3.73 	imes 10^3$ | 5.89 	imes 10 | | | |
| f28 | 2800 | $3.96 	imes 10^3$ | $1.11 	imes 10^2$ | $3.96 	imes 10^3$ | $1.35 	imes 10^2$ | | | |
| f29 | 2900 | $7.96 	imes 10^3$ | $5.34	imes10^2$ | $7.96 	imes 10^3$ | 5.11×10^2 | | | |
| f30 | 3000 | $3.04	imes10^5$ | $1.27 	imes 10^5$ | $3.18 	imes 10^5$ | $1.68 	imes 10^5$ | | | |

Table 5. Cont.

5. Discussion of Results

In the literature, we can find the hybrid firefly and particle swarm optimization algorithm for solving expensive computational problems [40], which was also used for the optimization of the CEC 2017 functions, and the performance of which was compared to another four optimization methods. The comparison was done in [40] with the combination of the hybrid firefly algorithm and particles swarm optimization (HFPSO), as well as with the 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, and for each case, the maximum number of evaluations of 500 was used for 10 dimensions (10D), and 1500 for 30 dimensions (30D).

The combination of the 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 [40], with respect to the other four optimization algorithms. For this reason, in Tables 6 and 7, 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 10 and 30 dimensions, respectively, show us that the dynamic stochastic fractal search method obtained, on average, better results in finding the global solutions of the functions. As previously mentioned, the results of the dynamic stochastic fractal search 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 HFPSO for 10 and 30 dimensions, respectively.

| Table 6. HFPSO vs. DSFS res | sults with 10 d | imensions |
|-----------------------------|-----------------|-----------|
|-----------------------------|-----------------|-----------|

| | | HFPSO [40] | | DSFS Type-1 Fuzzy Logic | | DSFS Type-2 Fuzzy Logic | |
|----------|-----|-------------------|----------------------|----------------------------|----------------------|----------------------------|----------------------|
| Function | fi | Mean | Std | Mean | Std | Mean | Std |
| f1 | 100 | $9.81 	imes 10^8$ | 1.01×10^2 | 1.01×10^2 | $4.87	imes10^{-1}$ | 1.01×10^2 | $7.85 	imes 10^{-1}$ |
| f2 | 200 | $4.91	imes10^8$ | $2.00 	imes 10^2$ | $2.00 	imes 10^2$ | 0.00 | $2.00 	imes 10^2$ | 0.00 |
| f3 | 300 | $5.96 	imes 10^3$ | 3.00×10^2 | 3.00×10^2 | $3.73 	imes 10^{-6}$ | 3.00×10^2 | $5.04	imes10^{-6}$ |
| f4 | 400 | 4.55 	imes 10 | 4.01×10^2 | 4.01×10^2 | $7.81	imes10^{-1}$ | 4.00×10^2 | $6.19	imes10^{-1}$ |
| f5 | 500 | 1.84 	imes 10 | 5.06×10^2 | 5.06×10^2 | 2.04 | $5.07 	imes 10^2$ | 2.64 |
| f6 | 600 | 1.35 	imes 10 | 6.00×10^{2} | 6.00×10^{2} | $7.21 	imes 10^{-8}$ | 6.00×10^{2} | $6.24 	imes 10^{-8}$ |
| f7 | 700 | 1.73 	imes 10 | $7.19	imes10^2$ | $7.19	imes10^2$ | 3.45 | $7.20 	imes 10^2$ | 3.32 |

| | | HEPSO [40] | | D | DSFS | | SFS |
|----------|------|--------------------|----------------------|--------------------|-----------------------|----------------------|------------------------|
| | | 11113 | | Type-1 F | Type-1 Fuzzy Logic | | uzzy Logic |
| Function | fi | Mean | Std | Mean | Std | Mean | Std |
| f8 | 800 | 1.44 	imes 10 | $8.07 	imes 10^2$ | $8.07 	imes 10^2$ | 3.04 | $8.07 	imes 10^2$ | 2.44 |
| f9 | 900 | 3.07×10^2 | 9.00×10^2 | 9.00×10^2 | 0.00 | 9.00×10^2 | 0.00 |
| f10 | 1000 | 3.79×10^2 | $1.40	imes10^3$ | $1.40 	imes 10^3$ | $1.70 	imes 10^2$ | $1.34	imes10^3$ | $1.31 	imes 10^2$ |
| f11 | 1100 | 5.24 	imes 10 | $1.10 	imes 10^3$ | $1.10 	imes 10^3$ | $9.31	imes10^{-1}$ | $1.10 	imes 10^3$ | 1.00 |
| f12 | 1200 | $4.13	imes10^6$ | 1.50×10^3 | $1.50 	imes 10^3$ | $1.04 	imes 10^2$ | 1.52×10^3 | $1.02 	imes 10^2$ |
| f13 | 1300 | $7.68 	imes 10^3$ | 1.31×10^3 | $1.31 	imes 10^3$ | 4.03 | $1.31 	imes 10^3$ | 4.14 |
| f14 | 1400 | $4.18	imes10^3$ | $1.40 	imes 10^3$ | $1.40 	imes 10^3$ | 1.69 | $1.40 	imes 10^3$ | 2.01 |
| f15 | 1500 | $2.37 	imes 10^4$ | 1.50×10^3 | 1.50×10^3 | $7.24	imes10^{-1}$ | 1.50×10^3 | $8.88 	imes 10^{-1}$ |
| f16 | 1600 | $1.59 	imes 10^2$ | 1.60×10^3 | 1.60×10^3 | $2.79	imes10^{-1}$ | 1.60×10^3 | $3.64 	imes 10^{-1}$ |
| f17 | 1700 | 8.40 	imes 10 | 1.70×10^3 | $1.70 	imes 10^3$ | 2.68 | $1.71 	imes 10^3$ | 4.41 |
| f18 | 1800 | $1.79 	imes 10^4$ | $1.81 	imes 10^3$ | $1.81 	imes 10^3$ | 2.56 | $1.81 	imes 10^3$ | 2.57 |
| f19 | 1900 | $3.83	imes10^4$ | $1.90 	imes 10^3$ | $1.90 	imes 10^3$ | $4.15	imes10^{-1}$ | $1.90 	imes 10^3$ | $4.45	imes10^{-1}$ |
| f20 | 2000 | $1.08 	imes 10^2$ | $2.00 	imes 10^3$ | 2.00×10^3 | $1.39	imes10^{-1}$ | 2.00×10^3 | $4.73 	imes 10^{-3}$ |
| f21 | 2100 | 4.78 	imes 10 | 2.23×10^3 | 2.23×10^3 | 5.08×10 | 2.24×10^3 | 5.40 	imes 10 |
| f22 | 2200 | $5.88 	imes 10^2$ | 2.29×10^{3} | 2.29×10^3 | 2.17×10 | $2.28 	imes 10^3$ | 3.93×10 |
| f23 | 2300 | 2.87 	imes 10 | 2.61×10^3 | 2.61×10^3 | 2.80 | $2.60 	imes 10^3$ | 5.61×10 |
| f24 | 2400 | $1.47 	imes 10^2$ | 2.61×10^{3} | 2.61×10^3 | 1.22×10^2 | 2.63×10^{3} | 1.14×10^2 |
| f25 | 2500 | 5.02×10 | $2.90 	imes 10^3$ | $2.90 	imes 10^3$ | 1.08 	imes 10 | $2.90 	imes 10^3$ | 1.15 	imes 10 |
| f26 | 2600 | 3.42×10^2 | 2.90×10^3 | 2.90×10^3 | $2.15 	imes 10^{-10}$ | 2.90×10^3 | 3.02×10^{-10} |
| f27 | 2700 | 3.94 	imes 10 | 3.09×10^3 | 3.09×10^3 | 1.97 | 3.09×10^3 | 2.14 |
| f28 | 2800 | $1.08 	imes 10^2$ | $3.09 	imes 10^3$ | $3.09 	imes 10^3$ | 4.20 	imes 10 | $3.11 	imes 10^3$ | 5.69×10 |
| f29 | 2900 | 9.40 	imes 10 | 3.16×10^3 | 3.16×10^3 | 1.01 	imes 10 | $3.16 	imes 10^3$ | 9.65 |
| f30 | 3000 | $3.75 	imes 10^6$ | 3.56×10^3 | 3.56×10^3 | 2.30×10^2 | $3.55 	imes 10^3$ | $1.19	imes 10^2$ |

| T .1.1. | | · · · · · |
|----------------|------|-----------|
| Table | b. (| .ont. |

| Table 7. HFPSO vs. 1 | DSFS results with 30 dimensions. |
|----------------------|----------------------------------|
| | |

| | | | DSFS | | DSFS | | |
|----------|------|----------------------|----------------------|--------------------|----------------------|--------------------|----------------------|
| | | nri 5 | U [40] | Type-1 Fuzzy Logic | | Type-2 Fuzzy Logic | |
| Function | fi | Mean | Std | Mean | Std | Mean | Std |
| f1 | 100 | $9.81 	imes 10^8$ | 1.01×10^2 | $3.49 	imes 10^3$ | 2.76×10^3 | $3.24 	imes 10^3$ | $2.74	imes10^3$ |
| f2 | 200 | $4.91 	imes 10^8$ | $2.00 	imes 10^2$ | $3.06	imes10^{16}$ | $7.11 	imes 10^{16}$ | $7.59	imes10^{17}$ | $3.98	imes10^{18}$ |
| f3 | 300 | 5.96×10^3 | 3.00×10^2 | $8.40	imes10^3$ | $3.95 	imes 10^3$ | $8.79 	imes 10^3$ | $4.60 	imes 10^3$ |
| f4 | 400 | 4.55 	imes 10 | 4.01×10^2 | $4.87 	imes 10^2$ | 3.56×10 | $4.80 	imes 10^2$ | 3.39 	imes 10 |
| f5 | 500 | 1.84 	imes 10 | $5.06 	imes 10^2$ | $6.11 	imes 10^2$ | 2.27 	imes 10 | $6.00 	imes 10^2$ | 2.02×10 |
| f6 | 600 | 1.35 	imes 10 | 6.00×10^{2} | $6.00 	imes 10^2$ | $1.36 	imes 10^{-2}$ | 6.00×10^2 | $1.13 	imes 10^{-2}$ |
| f7 | 700 | 1.73 	imes 10 | $7.19 	imes 10^2$ | $8.53 	imes 10^2$ | 1.68 	imes 10 | 8.60×10^2 | 1.70×10 |
| f8 | 800 | 1.44 	imes 10 | $8.07	imes10^2$ | $9.05	imes10^2$ | 2.12×10 | $9.06 	imes 10^2$ | 2.27 	imes 10 |
| f9 | 900 | 3.07×10^2 | $9.00 	imes 10^2$ | $9.01 	imes 10^2$ | $6.89	imes10^{-1}$ | 9.04×10^2 | 1.44 	imes 10 |
| f10 | 1000 | 3.79×10^{2} | $1.40 	imes 10^3$ | $6.21 	imes 10^3$ | $6.89	imes10^{-1}$ | $6.05 	imes 10^3$ | $5.03 	imes 10^2$ |
| f11 | 1100 | 5.24 	imes 10 | $1.10	imes10^3$ | $1.19	imes 10^3$ | 2.31 	imes 10 | $1.19	imes10^3$ | 2.90×10 |
| f12 | 1200 | $4.13	imes10^6$ | $1.50 	imes 10^3$ | $1.56 	imes 10^5$ | $1.03 	imes 10^5$ | $1.81 	imes 10^5$ | $1.48 	imes 10^5$ |
| f13 | 1300 | $7.68 	imes 10^3$ | $1.31 	imes 10^3$ | $3.56 	imes 10^3$ | $9.09 	imes 10^2$ | $4.01 	imes 10^3$ | $1.00 	imes 10^3$ |
| f14 | 1400 | $4.18	imes10^3$ | $1.40	imes10^3$ | $1.50	imes10^3$ | 1.03 	imes 10 | $1.50	imes10^3$ | 1.06 	imes 10 |
| f15 | 1500 | $2.37 	imes 10^4$ | $1.50 	imes 10^3$ | $1.70 	imes 10^3$ | 4.43 	imes 10 | $1.70 	imes 10^3$ | 3.95 	imes 10 |
| f16 | 1600 | $1.59 	imes 10^2$ | $1.60 	imes 10^3$ | $2.45 	imes 10^3$ | $2.66 	imes 10^2$ | $2.47 	imes 10^3$ | $2.14	imes10^2$ |
| f17 | 1700 | 8.40	imes10 | $1.70 	imes 10^3$ | $1.85	imes10^3$ | 6.65 	imes 10 | $1.86 	imes 10^3$ | 8.06 	imes 10 |
| f18 | 1800 | $1.79 	imes 10^4$ | $1.81 	imes 10^3$ | $2.87 	imes 10^3$ | $6.86 	imes 10^2$ | $2.69 	imes 10^3$ | $3.69 	imes 10^2$ |
| f19 | 1900 | $3.83	imes10^4$ | $1.90	imes10^3$ | $1.99	imes10^3$ | 1.88 	imes 10 | $1.99	imes10^3$ | 1.87 	imes 10 |
| f20 | 2000 | $1.08 	imes 10^2$ | $2.00 	imes 10^3$ | $2.39 	imes 10^3$ | 2.58 	imes 10 | $2.22 	imes 10^3$ | $1.12 	imes 10^2$ |
| f21 | 2100 | 4.78	imes10 | $2.23 	imes 10^3$ | $2.40	imes10^3$ | 2.28 	imes 10 | $2.39 	imes 10^3$ | 2.51 	imes 10 |
| f22 | 2200 | 5.88×10^2 | $2.29 	imes 10^3$ | $2.30 	imes 10^3$ | $1.50 	imes 10^{-2}$ | 2.30×10^3 | $7.44	imes10^{-3}$ |

| | | HFPSO [40] | | DSFS Type-1 Fuzzy Logic | | DSFS Type-2 Fuzzy Logic | |
|----------|------|--------------------|----------------------|----------------------------|--------------------|----------------------------|----------------------|
| Function | fi | Mean | Std | Mean | Std | Mean | Std |
| f23 | 2300 | 2.87×10 | 2.61×10^{3} | 2.74×10^3 | 2.86×10 | 2.73×10^{3} | 2.26×10 |
| f24 | 2400 | $1.47 	imes 10^2$ | 2.61×10^{3} | $2.91 	imes 10^3$ | 3.37×10 | 2.90×10^3 | 3.04 	imes 10 |
| f25 | 2500 | 5.02×10 | 2.90×10^3 | $2.89 	imes 10^3$ | 1.73 | $2.89 	imes 10^3$ | $9.76 	imes 10^{-1}$ |
| f26 | 2600 | 3.42×10^2 | $2.90 	imes 10^3$ | $4.28 	imes 10^3$ | 5.50×10^2 | $4.22 	imes 10^3$ | $6.59 	imes 10^2$ |
| f27 | 2700 | 3.94×10 | $3.09 	imes 10^3$ | 3.22×10^3 | 8.08 | 3.22×10^3 | 7.37 |
| f28 | 2800 | 1.08×10^2 | 3.09×10^3 | 3.21×10^3 | 1.25×10 | 3.21×10^3 | 1.27×10 |
| f29 | 2900 | 9.40 	imes 10 | $3.16 	imes 10^3$ | $3.63 	imes 10^3$ | $1.06 	imes 10^2$ | $3.59 	imes 10^3$ | $1.13 	imes 10^2$ |
| f30 | 3000 | $3.75 	imes 10^6$ | $3.56 	imes 10^3$ | $1.32 	imes 10^4$ | 3.29×10^3 | $1.51 	imes 10^4$ | $5.55 	imes 10^3$ |

Table 7. Cont.









CEC 2017 Results with 30 dimensions

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

On the vertical axis, we can find the values obtained with the DSFS and HFPSO methods, and the horizontal axis shows the corresponding functions. The DSFS approach, indicated by the blue line, has better efficiency than the HFPSO (orange line) because the values are closer to the global optima, as seen in Figure 3.

Figure 4 illustrates the results obtained with 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 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, and 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, 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 as when using lower dimensions, where the efficiency of the algorithms is much better for each of the functions.

To determine which of the HFPSO or DSFS provided the closest to the 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:

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

DEEC

where $\overline{x}_1 - \overline{x}_2$ is the observed difference in the mean values of the methods, $\mu_1 - \mu_2$ is the expected difference in the mean values of the methods, and $\sigma_{\overline{x}_1 - \overline{x}_2}$ is the standard error of the differences.

As can be seen in Table 8, 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 the functions with 10 dimensions (bold type indicates the best values).

| | | HFPS | O [40] | Type-1 Fuzzy Logic | | |
|----------|------|-------------------|--------------------|----------------------|----------------------|-------------------|
| Function | fi | Mean | Std | Mean | Std | Z |
| f1 | 100 | $9.81 	imes 10^8$ | 1.01×10^2 | 1.01×10^2 | $4.87 	imes 10^{-1}$ | $3.76 	imes 10^7$ |
| f2 | 200 | $4.91 	imes 10^8$ | $2.00 	imes 10^2$ | 2.00×10^2 | 0.00 | $9.51 	imes 10^6$ |
| f3 | 300 | $5.96	imes10^3$ | $3.00 	imes 10^2$ | 3.00×10^2 | $3.73	imes10^{-6}$ | 7.31×10 |
| f4 | 400 | 4.55 	imes 10 | $4.01 	imes 10^2$ | 4.01×10^2 | $7.81	imes10^{-1}$ | -3.43 |
| f5 | 500 | 1.84 	imes 10 | $5.06 	imes 10^2$ | 5.06×10^2 | 2.04 | -3.73 |
| f6 | 600 | 1.35 	imes 10 | $6.00 	imes 10^2$ | 6.00×10^{2} | $7.21	imes10^{-8}$ | -3.79 |
| f7 | 700 | 1.73×10 | $7.19 	imes 10^2$ | $7.19 	imes 10^2$ | 3.45 | -3.78 |
| f8 | 800 | 1.44 	imes 10 | $8.07 	imes 10^2$ | $8.07 	imes 10^2$ | 3.04 | -3.80 |
| f9 | 900 | $3.07 	imes 10^2$ | $9.00 	imes 10^2$ | 9.00×10^{2} | 0.00 | -2.55 |
| f10 | 1000 | $3.79 	imes 10^2$ | $1.40 	imes 10^3$ | $1.40 	imes 10^3$ | $1.70 	imes 10^2$ | -2.82 |
| f11 | 1100 | 5.24 	imes 10 | $1.10 	imes 10^3$ | $1.10 	imes 10^3$ | $9.31	imes10^{-1}$ | -3.69 |
| f12 | 1200 | $4.13	imes10^6$ | $1.50 	imes 10^3$ | $1.50 	imes 10^3$ | $1.04 	imes 10^2$ | $1.07 	imes 10^4$ |
| f13 | 1300 | $7.68 	imes 10^3$ | $1.31 	imes 10^3$ | 1.31×10^3 | 4.03 | 1.88 	imes 10 |
| f14 | 1400 | $4.18	imes10^3$ | $1.40 	imes 10^3$ | $1.40 	imes 10^3$ | 1.69 | 7.69 |
| f15 | 1500 | $2.37 	imes 10^4$ | $1.50 	imes 10^3$ | $1.50 	imes 10^3$ | $7.24	imes10^{-1}$ | 5.73 	imes 10 |
| f16 | 1600 | $1.59 	imes 10^2$ | $1.60 	imes 10^3$ | 1.60×10^3 | $2.79	imes10^{-1}$ | -3.49 |
| f17 | 1700 | 8.40 	imes 10 | $1.70 	imes 10^3$ | $1.70 	imes 10^3$ | 2.68 | -3.68 |
| f18 | 1800 | $1.79 	imes 10^4$ | $1.81 	imes 10^3$ | $1.81 	imes 10^3$ | 2.56 | 3.44×10 |
| f19 | 1900 | $3.83	imes10^4$ | $1.90 	imes 10^3$ | $1.90 	imes 10^3$ | $4.15	imes10^{-1}$ | 7.42×10 |
| f20 | 2000 | $1.08 	imes 10^2$ | $2.00 	imes 10^3$ | $2.00 	imes 10^3$ | $1.39	imes10^{-1}$ | -3.66 |
| f21 | 2100 | 4.78 	imes 10 | 2.23×10^3 | 2.23×10^3 | 5.08 	imes 10 | -3.79 |

 Table 8. Z-test for 10 dimensions.

| | | HFPS | O [40] | Г Туре-1 І | | |
|----------|------|-------------------|--------------------|--------------------|----------------------|--------------------|
| Function | fi | Mean | Std | Mean | Std | Z |
| f22 | 2200 | $5.88 	imes 10^2$ | $2.29 	imes 10^3$ | $2.29 	imes 10^3$ | 2.17 	imes 10 | -2.88 |
| f23 | 2300 | 2.87 	imes 10 | $2.61 	imes 10^3$ | $2.61 	imes 10^3$ | 2.80 | -3.83 |
| f24 | 2400 | $1.47 	imes 10^2$ | $2.61 	imes 10^3$ | $2.61 	imes 10^3$ | $1.22 	imes 10^2$ | -3.65 |
| f25 | 2500 | 5.02×10 | $2.90 	imes 10^3$ | $2.90 	imes 10^3$ | 1.08 	imes 10 | -3.81 |
| f26 | 2600 | $3.42 	imes 10^2$ | $2.90 	imes 10^3$ | $2.90 	imes 10^3$ | $2.15	imes10^{-10}$ | -3.42 |
| f27 | 2700 | 3.94 	imes 10 | $3.09 	imes 10^3$ | 3.09×10^3 | 1.97 | -3.82 |
| f28 | 2800 | $1.08 	imes 10^2$ | $3.09 	imes 10^3$ | $3.09 	imes 10^3$ | 4.20 	imes 10 | -3.74 |
| f29 | 2900 | 9.40 	imes 10 | $3.16 	imes 10^3$ | $3.16 	imes 10^3$ | 1.01 	imes 10 | -3.76 |
| f30 | 3000 | $3.75 	imes 10^6$ | 3.56×10^3 | 3.56×10^3 | 2.30×10^{2} | 4.08×10^3 |

Table 8. Cont.

Table 9 describes in Column 7 the results when applying the Z-test to the methods, compared to DSFS with type-1 fuzzy logic vs. HFPSO using 30 dimensions. Again, the proposed method is significantly better. For example, in Row 2, for the f2 function, the Z value provides sufficient statistical evidence that the proposed DSFS method is better.

Table 9. Results Z-test for 30 dimensions.

| | | HFPS | HFPSO [40] DSFS Type-1 Fuzzy Logic | | | |
|----------|------|--------------------|---------------------------------------|----------------------|----------------------|--------------------|
| Function | fi | Mean | Std | Mean | Std | z |
| f1 | 100 | $9.81 	imes 10^8$ | $1.01 	imes 10^2$ | $3.49 	imes 10^3$ | 2.76×10^3 | $1.95 	imes 10^6$ |
| f2 | 200 | $4.91	imes10^8$ | $2.00 	imes 10^2$ | $3.06	imes10^{16}$ | $7.11	imes10^{16}$ | -2.36 |
| f3 | 300 | $5.96 	imes 10^3$ | $3.00 	imes 10^2$ | $8.40 	imes 10^3$ | $3.95 	imes 10^3$ | -3.37 |
| f4 | 400 | 4.55 	imes 10 | $4.01 	imes 10^2$ | $4.87 	imes 10^2$ | 3.56 	imes 10 | -6.01 |
| f5 | 500 | 1.84 	imes 10 | $5.06 	imes 10^2$ | 6.11×10^{2} | 2.27×10 | -6.41 |
| f6 | 600 | 1.35×10 | $6.00 	imes 10^2$ | 6.00×10^2 | $1.36 	imes 10^{-2}$ | -5.35 |
| f7 | 700 | 1.73×10 | $7.19	imes10^2$ | $8.53 	imes 10^2$ | 1.68 	imes 10 | -6.36 |
| f8 | 800 | 1.44×10 | $8.07 	imes 10^2$ | $9.05 	imes 10^2$ | 2.12×10 | -6.04 |
| f9 | 900 | $3.07 	imes 10^2$ | $9.00 	imes 10^2$ | 9.01×10^{2} | $6.89	imes10^{-1}$ | -3.61 |
| f10 | 1000 | $3.79 	imes 10^2$ | $1.40	imes10^3$ | 6.21×10^3 | $6.89	imes10^{-1}$ | -2.28×10 |
| f11 | 1100 | 5.24 	imes 10 | $1.10 	imes 10^3$ | $1.19 	imes 10^3$ | 2.31×10 | -5.66 |
| f12 | 1200 | $4.13	imes10^6$ | $1.50 	imes 10^3$ | 1.56×10^5 | $1.03	imes10^5$ | 2.11×10^2 |
| f13 | 1300 | $7.68 	imes 10^3$ | $1.31 	imes 10^3$ | $3.56 	imes 10^3$ | $9.09	imes10^2$ | 1.42×10 |
| f14 | 1400 | $4.18 	imes 10^3$ | $1.40 	imes 10^3$ | 1.50×10^3 | 1.03 	imes 10 | 1.05 	imes 10 |
| f15 | 1500 | $2.37 	imes 10^4$ | $1.50 	imes 10^3$ | 1.70×10^3 | 4.43 	imes 10 | 8.03 	imes 10 |
| f16 | 1600 | $1.59 	imes 10^2$ | $1.60 	imes 10^3$ | $2.45 	imes 10^3$ | $2.66 	imes 10^2$ | -7.74 |
| f17 | 1700 | 8.40 	imes 10 | $1.70 	imes 10^3$ | $1.85 	imes 10^3$ | 6.65 	imes 10 | -5.69 |
| f18 | 1800 | $1.79 	imes 10^4$ | $1.81 	imes 10^3$ | $2.87 	imes 10^3$ | $6.86 	imes 10^2$ | 4.25 	imes 10 |
| f19 | 1900 | $3.83	imes10^4$ | $1.90 	imes 10^3$ | $1.99 	imes 10^3$ | 1.88 	imes 10 | $1.05 	imes 10^2$ |
| f20 | 2000 | $1.08 	imes 10^2$ | $2.00 	imes 10^3$ | 2.39×10^3 | 2.58 	imes 10 | -6.25 |
| f21 | 2100 | 4.78 	imes 10 | $2.23 	imes 10^3$ | $2.40 	imes 10^3$ | 2.28 	imes 10 | -5.78 |
| f22 | 2200 | $5.88 	imes 10^2$ | $2.29 	imes 10^3$ | $2.30 	imes 10^3$ | $1.50	imes10^{-2}$ | -4.09 |
| f23 | 2300 | 2.87 	imes 10 | $2.61 	imes 10^3$ | $2.74 	imes 10^3$ | 2.86 	imes 10 | -5.69 |
| f24 | 2400 | $1.47 	imes 10^2$ | $2.61 	imes 10^3$ | 2.91×10^3 | 3.37 	imes 10 | -5.80 |
| f25 | 2500 | 5.02×10 | $2.90 	imes 10^3$ | $2.89 	imes 10^3$ | 1.73 | -5.36 |
| f26 | 2600 | 3.42×10^2 | $2.90 	imes 10^3$ | $4.28 	imes 10^3$ | $5.50 	imes 10^2$ | -7.31 |
| f27 | 2700 | 3.94 	imes 10 | $3.09 	imes 10^3$ | 3.22×10^{3} | 8.08 | -5.64 |
| f28 | 2800 | $1.08 	imes 10^2$ | $3.09 	imes 10^3$ | 3.21×10^3 | 1.25 	imes 10 | -5.50 |
| f29 | 2900 | 9.40 	imes 10 | $3.16 	imes 10^3$ | $3.63 	imes 10^3$ | $1.06 	imes 10^2$ | -6.13 |
| f30 | 3000 | $3.75 	imes 10^6$ | $3.56 	imes 10^3$ | $1.32 	imes 10^4$ | 3.29×10^3 | 4.22×10^3 |

6. Conclusions

In conclusion, we can propose efficient stochastic methods, such as metaheuristics, that include basic algorithms for global stochastic optimization, such as the random search, which helps with 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 parts of the 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.

The experimentation was carried out first with the comparison of the results of the dynamic stochastic fractal search method, adjusted with both kinds of fuzzy logic, where we found that the values were very similar for all the CEC 2017 evaluation functions. After that, a comparison with the combination of the hybrid firefly algorithm and particle swarm optimization (HFPSO) was also done, showing a significant statistical advantage for the proposed DSFS method of this paper. In addition, since HFPSO was better than the FA, PSO, HPSOFF, and FFPSO [39], the proposed DSFS is also better than these five metaheuristic optimization algorithms.

It is shown that the improvement applied to the method was satisfactory because of 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 fuzzy inference controller. 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 was developed in this study. In future work, we plan to apply the proposed DFSF to real-world problems in different areas of application.

Author Contributions: Conceptualization, M.L.L. and O.C.; methodology, P.M.; software, F.V.; validation, J.S., M.L.L. and O.C.; formal analysis, J.S.; investigation, M.L.L.; resources, P.M.; data curation, M.L.L.; writing—original draft preparation, M.L.L.; writing—review and editing, O.C.; visualization, F.V.; supervision, O.C.; project administration, P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Yang, X.-S. Nature-Inspired Metaheuristic Algorithms; Luniver Press: Cambridge, UK, 2010.
- 2. Holland, J.H. Genetic Algorithms understand Genetic Algorithms. Sci. Am. 1992, 267, 66–73. [CrossRef]
- 3. Garg, H. 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: Seri Iskandar, Malaysia, 2015; pp. 620–654.
- 4. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
- 5. Glover, F.; Laguna, M. *Tabu Search;* Springer: New York, NY, USA, 1997.
- 6. Gallego, R.A.; Romero, R.; Monticelli, A.J. Tabu search algorithm for network synthesis. *IEEE Trans. Power Syst.* 2000, 15, 490–495. [CrossRef]
- Liu, B.; Wang, L.; Jin, Y.H.; Tang, F.; Huang, D.X. Improved particle swarm optimization combined with chaos. *Chaos Solitons Fractals* 2005, 25, 1261–1271. [CrossRef]
- Alatas, B.; Akin, E.; Ozer, A.B. Chaos embedded particle swarm optimization algorithms. *Chaos Solitons Fractals* 2009, 40, 1715–1734. [CrossRef]

- 9. Jiao, B.; Lian, Z.; Gu, X. A dynamic inertia weight particle swarm optimization algorithm. *Chaos Solitons Fractals* **2008**, 37, 698–705. [CrossRef]
- 10. Yang, D.; Li, G.; Cheng, G. On the efficiency of chaos optimization algorithms for global optimization. *Chaos Solitons Fractals* **2007**, 34, 1366–1375. [CrossRef]
- 11. Misaghi, M.; Yaghoobi, M. Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller. *J. Comput. Des. Eng.* **2019**, *6*, 284–295. [CrossRef]
- 12. Chen, B.; Liu, X.; Tong, S. Adaptive fuzzy approach to control unified chaotic systems. *Chaos Solitons Fractals* 2007, 34, 1180–1187. [CrossRef]
- 13. Lagunes, M.L.; Castillo, O.; Valdez, F.; Soria, J. Comparative Study of Fuzzy Controller Optimization with Dynamic Parameter Adjustment Based on Type 1 and Type 2 Fuzzy Logic. *Int. Fuzzy Syst. Assoc. World Congr.* **2019**, *1000*, 296–305.
- Lagunes, M.L.; Castillo, O.; Valdez, F.; Soria, J.; 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: Berlin/Heidelberg, Germany, 2018; pp. 569–579.
- 15. Amador-Angulo, L.; 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: Berlin/Heidelberg, Germany, 2017; pp. 551–571.
- 16. Lagunes, M.L.; Castillo, O.; Valdez, F.; Soria, J. Comparison of Fuzzy Controller Optimization with Dynamic Parameter Adjustment Based on of Type-1 and Type-2 Fuzzy Logic. *Hybrid Intell. Syst. Control Pattern Recognit. Med.* **2020**, *827*, 47–56.
- 17. Pérez, J.; Valdez, F.; 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: Berlin/Heidelberg, Germany, 2017; pp. 343–355.
- Bernal, E.; Castillo, O.; Soria, J.; Valdez, F.; Melin, P. A variant to the dynamic adaptation of parameters in galactic swarm optimization using a fuzzy logic augmentation. In Proceedings of the 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7.
- 19. Sánchez, D.; Melin, P.; Castillo, O. Optimization of modular granular neural networks using a firefly algorithm for human recognition. *Eng. Appl. Artif. Intell.* **2017**, *64*, 172–186. [CrossRef]
- 20. Mandelbrot, B. The Fractal Geometry of Nature; W. H. Freeman and Company: San Francisco, CA, USA, 1982; Volume 1.
- 21. Barnsley, M.F.; Demko, S. Iterated function systems and the global construction of fractals. *Proc. R. Soc. Lond. Ser. A Math. Phys. Sci.* **1985**, 399, 243–275.
- 22. Grassberger, P.; Procaccia, I. Characterization of strange attractors. Phys. Rev. Lett. 1983, 50, 346–349. [CrossRef]
- 23. Prusinkiewicz, P. Graphical applications of L-systems. Proc. Graph. Interface 1986, 86, 247–253.
- 24. Rushton, B. Subdivision rules for all Gromov hyperbolic groups. arXiv 2017, arXiv:1708.02366.
- 25. Falconer, K.J. Random fractals. Math. Proc. Camb. Philos. Soc. 1986, 100, 559–582. [CrossRef]
- 26. Khalilpourazari, S.; Khalilpourazary, S. A Robust Stochastic Fractal Search approach for optimization of the surface grinding process. *Swarm Evol. Comput.* **2018**, *38*, 173–186. [CrossRef]
- Mellal, M.A.; Zio, E. A penalty guided stochastic fractal search approach for system reliability optimization. *Reliab. Eng. Syst. Saf.* 2016, 152, 213–227. [CrossRef]
- 28. Salimi, H. Stochastic Fractal Search: A powerful metaheuristic algorithm. Knowl. Based Syst. 2015, 75, 1–18. [CrossRef]
- 29. Khalilpourazari, S.; Naderi, B.; Khalilpourazary, S. Multi-Objective Stochastic Fractal Search: A powerful algorithm for solving complex multi-objective optimization problems. *Soft Comput.* **2020**, *24*, 3037–3066. [CrossRef]
- 30. Aras, S.; Gedikli, E.; Kahraman, H.T. A novel stochastic fractal search algorithm with fitness-Distance balance for global numerical optimization. *Swarm Evol. Comput.* 2021, *61*, 100821. [CrossRef]
- 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**, *30*, 1991–2002. [CrossRef]
- 32. Alomoush, M.I.; Oweis, Z.B. Environmental-economic dispatch using stochastic fractal search algorithm. *Int. Trans. Electr. Energy Syst.* **2018**, *28*, e2530. [CrossRef]
- Miramontes, I.; Melin, P.; 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: Berlin/Heidelberg, Germany, 2020; Volume 827, pp. 219–231.
- 34. Miramontes, I.; Melin, P.; Prado-Arechiga, G. Fuzzy System for Classification of Nocturnal Blood Pressure Profile and Its Optimization with the Crow Search Algorithm. *Adv. Intell. Syst. Comput.* **2021**, *1222*, 23–34.
- 35. Guzmán, J.; Miramontes, I.; Melin, P.; Prado-Arechiga, G. Optimal Genetic Design of Type-1 and Interval Type-2 Fuzzy Systems for Blood Pressure Level Classification. *Axioms* 2019, *8*, 8. [CrossRef]
- 36. Carvajal, O.; Melin, P.; Miramontes, I.; 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**, *97*, 104069. [CrossRef]
- 37. Miramontes, I.; Guzman, J.C.; Melin, P.; Prado-Arechiga, G. Optimal design of interval type-2 fuzzy heart rate level classification systems using the bird swarm algorithm. *Algorithms* **2018**, *11*, 206. [CrossRef]
- 38. Biedrzycki, R. A Version of IPOP-CMA-ES Algorithm with Midpoint for CEC 2017 Single Objective Bound Constrained Problems. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Donostia, Spain, 5–8 June 2017; pp. 1489–1494.

- 39. Awad, N.H.; Ali, M.; Liang, J.; Qu, B.; Suganthan, P.; Definitions, P. *Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Obejective Real-Parameter Numerical Optimization;* Technology Report; Nanyang Technological University: Singapore; Jordan Zhengzhou University: Zhengzhou, China, 2016.
- 40. Aydilek, İ.B. A Hybrid Firefly and Particle Swarm Optimization Algorithm for Computationally Expensive Numerical Problems. *Appl. Soft Comput.* **2018**, *66*, 232–249. [CrossRef]