algorithms

MDPI

Article

Multi-Metaheuristic Competitive Model for Optimization of Fuzzy Controllers

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

Tijuana Institute of Technology, c.p 22379 Tijuana, Mexico; marylu.lara@tectijuana.edu.mx (M.L.L.); fevrier@tectijuana.mx (F.V.); jsoria57@gmail.com (J.S.)

* Correspondence: ocastillo@tectijuana.mx; Tel.: +52-664-623-6318

Received: 26 March 2019; Accepted: 23 April 2019; Published: 28 April 2019

check for updates

Abstract: This article describes an optimization methodology based on a model of competitiveness between different metaheuristic methods. The main contribution is a strategy to dynamically find the algorithm that obtains the best result based on the competitiveness of methods to solve a specific problem using different performance metrics depending on the problem. The algorithms used in the preliminary tests are: the firefly algorithm (FA), which is inspired by blinking fireflies; wind-driven optimization (WDO), which is inspired by the movement of the wind in the atmosphere, and in which the positions and velocities of the wind packages are updated; and finally, drone squadron optimization (DSO)—the inspiration for this method is new and interesting—based on artifacts, where drones have a command center that sends information to individual drones and updates their software to optimize the objective function. The proposed model helps discover the best method to solve a specific problem, and also reduces the time that it takes to search for methods before finding the one that obtains the most satisfactory results. The main idea is that with this competitiveness approach, methods are tested at the same time until the best one to solve the problem in question is found. As preliminary tests of the model, the optimization of the benchmark mathematical functions and membership functions of a fuzzy controller of an autonomous mobile robot was used.

Keywords: firefly algorithm; drone squadron optimization; wind-driven optimization; optimization; fuzzy controller

1. Introduction

Metaheuristics, fuzzy logic, and artificial neural networks are models that are related to artificial intelligence and are currently used to carry out optimization processes, solve control problems with uncertainty, and perform predictions, among other problems found in industry, medicine, scientific areas, etc. For example, in [1], a model for the development of a human resources portfolio that uses a neuro-fuzzy approach is presented. Meanwhile, systems that are created using type-2 adaptive neural networks for processing parameters are developed in [2]. Some other applications describe the development of models based on fuzzy logic and artificial neural networks for prediction as in [3]. Also, in [4], neuro-fuzzy adaptive inference systems are used for modeling complex systems. In addition, [5] developed an adaptive neuro-fuzzy inference guidance system that provides instructions to drivers based on optimal route solutions. These neuro-fuzzy applications are being applied more and more as is the case of [6], where a neuro-fuzzy inference system is constructed for the evaluation of costs and risks in the selection of multiple objectives of transport routes of hazardous materials in a road network. Finally, Pamučar et al. [7] mentioned the use of fuzzy logic for modeling approaches for railroad administration support. Most metaheuristics are inspired by nature, where they take the main characteristics of survival of some individuals and become a new method, which is developed to solve real problems that a simple heuristic sometimes cannot solve. As an example, the ant colony algorithm in [8,9], and Dorigo et al. [10] also describes applications of this method, giving good results to solve optimization problems, finding routes or paths that lead to find the best solution in a specific search space, which is based on how the ants find the shortest path from their anthill to their food. Another example is the genetic algorithm, which is also a widely used metaheuristic and is inspired by the biological evolution developed by John Henry Holland in [11,12], where the algorithm is based on the idea of the survival of the fittest, going from generation in generation. In this case, each generation has a best individual, which is considered the one that has a better performance to find a good solution to a problem. Some examples of the use of the aforementioned algorithms are the following. In [13], a new hybrid Gravitational Search Algorithm-Genetic Algorithm (GSA-GA) algorithm is presented for the restriction of nonlinear optimization problems with mixed variables. A hydrothermal system is also developed for storage by pumping that incorporates solar units using particle swarm optimization as described in [14]. Another example of the use of metaheuristics is the hybrid technique called Particle Swarm Optimization-Genetic Algorithm PSO-GA to solve restricted optimization problems [15]; for multi-objective problems, particle swarm optimization has been used as explained in [16]. In addition, Garg [17] illustrates a problem of structural engineering design optimization with limitations of nonlinear resources using the artificial bee colony algorithm. In addition, in the control area, metaheuristics are combined with fuzzy logic to solve optimization problems, such as the optimization of membership functions to improve the behavior of a fuzzy controller. For example, in [18], the firefly algorithm is used for the optimization of a fuzzy controller of a standalone mobile robot, which performs dynamic adjustment of the randomness parameter of the method. Also in [19], fuzzy dynamic adjustment is performed for optimization using galactic swarm optimization, among others, and [20–22] also describe applications of the method. This article proposes the development of a multi-metaheuristic competitive model using the firefly algorithm (FA), the wind-driven particle optimization algorithm (WDO), and the drone squad optimization (DSO) algorithm, where each of them competes with each other to demonstrate which of them is the best at solving optimization problems. The question to answer in this work is: could a multi-metaheuristic competitive method improve the performance of a fuzzy control? The algorithms chosen for this model were selected because they proved to give good results in optimization problems according to the state of the art and experimentation. The main contribution in this work is the development of a competitive model that will serve as a methodology to solve optimization problems. The motivation in this work arises with the concern to solve the need that exists regarding having an optimization methodology that helps us evaluate different optimization algorithms by means of competitiveness between methods, and thus solve the problems of optimization that arise in an application area. The objective is to find the method that provides the best results in a particular optimization problem. Preliminary tests were performed with unimodal, multimodal, multimodal benchmark functions of fixed dimension, in addition to the optimization of a fuzzy controller of a robot that follows the line. This paper is organized as follows: Section 2 describes the background of the methods used, Section 3 presents the proposed methodology and shows the problems to be solved, Section 4 describes the results, Section 5 describes the discussion, and finally Section 6 show the conclusions.

2. Background on FA, WDO, DSO, and Fuzzy Logic

This section describes the characteristics and behavior of the selected methods for the development of the proposed model. These methods were chosen from among many existing optimization algorithms, because these have been shown to provide good results in solving optimization problems, in addition to being relatively new and innovative. For this reason, this paper also intends to give them the opportunity to test their potential in optimization performance.

2.1. Firefly Algorithm (FA)

The firefly algorithm (FA) is inspired on the flickering fireflies, and was developed by Yang in 2008 in [23], and in [24], and applications of the method are presented in [25]. This method is a metaheuristic that is developed based on three important rules.

1. All fireflies are the same.
2. Less bright fireflies will move toward the more bright fireflies.
3. The search space is given by the objective function.

In Equation (1), the attractiveness of a firefly is proportional to other fireflies, and the variation of β and the distance r are given as follows.

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{1}$$

where β_0 is the initial attractiveness at $r = 0$, e represents the basis of the natural logarithms, γ determines the variation of attractiveness as the distance between the fireflies increases, and r is the distance between each two fireflies.

The movement of a firefly i that is attracted by another more attractive firefly is determined by Equation (2):

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha_t \varepsilon_i^t \tag{2}$$

The current position is represented by the term x_i^t, the second term determines the initial attraction of the firefly β_0, γ is the absorption coefficient, and r is the Euclidean distance between the positions of the firefly i and the firefly j. The last term handles the exploration, where α is the parameter that controls how much randomness the firefly is allowed to have in its movement, and is a vector containing random numbers drawn from a Gaussian distribution or uniform distribution at time t.

In Equation (3), the r variable represents the distance between two fireflies, i and j.

$$r_{ij} = \sqrt{\sum_{k=1}^{d}\left(x_{i,k} - x_{j,k}\right)^2} \tag{3}$$

where $x_{i,k}$ is the kth component of the spatial coordinate, x_i is the position of a firefly, and r_{ij} is the Euclidean distance between two fireflies i and j.

2.2. Wind-Driven Optimization (WDO)

WDO is relatively new a metaheuristic inspired by the movement of the wind in the atmosphere, and has been used to solve multidimensional problems. It was developed by [26,27]. In this case, air packages represent the population in which the position and speed of their movement are updated in each iteration.

$$u_{t+1}^i = (1-)u_t^i - gx_t^i\left(RT\left|\frac{1}{r} - 1\right|\left(x_{opt} - x_t^i\right)\right) + \left(\frac{cu_t^{other\ dim}}{r}\right) \tag{4}$$

Equation (4) represents the new velocity that consists of a first term where one is the maximum pressure, α is the coefficient of friction, and u_t^i is the current velocity. The second term is composed by g, which corresponds to the gravity, while x_t^i is the current position, RT represents the universal gas constant and temperature respectively, r represents the range of the air pack where all the air particles are classified in descending order with respect to their pressure, and x_{opt} is the best global position, where the Coriolis constant is c.

$$x_{t+1}^i = x_t^i + u_{t+1}^i \tag{5}$$

and Equation (5) represents the next position in the WDO.

2.3. Drone Squadron Optimization (DSO)

In the DSO algorithm, the drones can navigate autonomously, and this method was originally developed in [28] and [29], and Yalcin et al. [30] describes some applications of the method. The sensors in DSO can communicate over long distances and use solar energy, and one of the most important features is that they can be updated not only regarding hardware, but also when changing their software.

The DSO algorithm consists of the simulation of a drone squadron with different equipment and a command center. The command center uses information collected by the drones to perform two operations:

1.- Maintain partial control of the search.
2.- Develop new control software for drones.

In general, the DSO consists of:

- Coordinates: representing a numerical solution.
- Scan the landscape: calculate the objective function and obtain the fitness value.
- Firmware: distinct rules/configurations to evolve the population.
- Team: group of agents that share a firmware, but operate on possibly different data.
- Squadron: group of teams with different firmware.

The following Figure 1 shows the DSO search cycle.

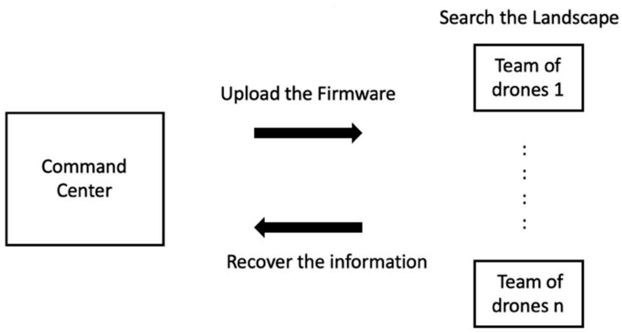

Figure 1. Drone Squadron Optimization (DSO) search cycle.

DSO has two main equations that establish the rules of drone movement as shown below in Equations (6) and (7):

$$P = Departure + Offset() \tag{6}$$

$$TC = calculate(P) \tag{7}$$

where P is the complete perturbation formula that should be calculated to return the trial coordinates, the variable *Departure* is a particular coordinate, $Offset$ is a function that returns the actual perturbation movement, and TC generates new trial coordinates through perturbation.

$$violation_{team} = \sum_{drone=1}^{N} \sum_{j=1}^{D} \left\{ \begin{array}{c} |TmC - UB_J| \\ + \\ |LB_j - TmC| \end{array} \right. \tag{8}$$

Equation (8) is applied when a drone commits a violation when leaving the limits of the search space, where $violation_{team}$ is equal to the infraction, TmC represents the equipment coordinates, UB_J and LB_j correspond to the upper and lower limits, respectively, D is equal to the number of dimensions of the problem, and finally N represents the number of drones per team.

To better explain the difference between the used methods, a comparison is made among them, as illustrated in Table 1, with respect to one of the most used metaheuristics: particle swarm optimization

(PSO) as developed by Kennedy and Eberhart in [31,32]; applications of the method are also described in [33]. PSO is a bio-inspired method in nature, to be more specific in the behavior of the birds, where each one represents a particle. Considering that the WDO algorithm is inspired by PSO, the following comparison table is made, and it should be noted that PSO is not currently used in this methodology, but rather only used in this table for comparison purposes.

Table 1. Comparison of the methods. DSO: drone squadron optimization; FA: firefly algorithm; PSO: particle swarm optimization; WDO: wind-driven optimization.

Characteristics	PSO	FA	WDO	DSO
Population	Particle	Firefly	Air package	Drone
New speed	v_{k+1}^{i}	-	u_{t+1}^{i}	-
Current speed	v_{k}^{i}	-	u_{t}^{i}	-
Actual position	x_i	τ	x_{t}^{i}	
Next position	x_{k+1}^{i}	x_{i}^{t+1}	x_{t+1}^{i}	
Better experience	p_{k}^{i}	-		
Best group experience	p_{k}^{g}	-		
Increase	k	t		
Uniform random numbers between 0 and 1	r_1 , r_2	-		
Cognitive parameter	c_1			
Social parameter	c_2			

2.4. Fuzzy Logic

Fuzzy logic was originally proposed by Zadeh in [34,35], and Zadeh [36] also describes recent applications of the method. It is a tool that makes life more comfortable today, because it solves problems that previously seemed to have no satisfactory solution. This is thanks to the use of linguistic variables that can model human ways of thinking, and it is much easier to program artifacts that can be manipulated using the logic of human reasoning. Fuzzy logic is combined with other artificial intelligence techniques, and this combination is able to achieve something that once seemed very distant: autonomous mobile robots. These include those robots that currently help with home duties, and even open heart operations in hospitals, but that's not all: in the world of industry, robots also play a very important role in the manufacturing processes. In order for these robots to have that efficiency in their work environment, it is very important that their software and hardware are constantly being updated and optimized, as argued in [37].

We propose a fuzzy logic controller that is applied to create a soft response instead of a traditional hard logical response. Other combinations and metaheuristics have been also used in conjunction with fuzzy logic in robotics as in [38,39], and El Ferik et al. [40] also describes applications of the method.

3. Proposed Methodology

The methodology consists of creating a competitive model based on a set of metaheuristics, and the motivation for this proposal was born with the aim of streamlining the search processes for optimization methods, because many times in research when it comes to optimizing a specific problem, time is lost experimenting with one metaheuristic after another, until finding the one that adapts better to the problem to have an optimization with a satisfactory result. The general idea of this proposal is to have a series of optimization methods, which receive an input (specific problem) to be processed, and thus optimized. The method that produces a better result than the others in the competition will show that it is the best to optimize that problem, and a metric is used to evaluate the particular results of the problem. The motivation is to develop an optimization methodology where several metaheuristics are used, thus evaluating their performance in optimizing and solving a particular problem.

Figure 2 shows the data flow of the proposed methodology. Step 1 is the input of the problem that will be optimized, which is processed by the methods in steps 2–5. Then, in step 6, the simulation to obtain the results is performed; in step 7, we show the ranking of errors. Step 8 illustrates the best obtained error, in this case of minimization using the mean square error (MSE), where important data is obtained by the method, such as time, iteration, and error. In step 9, the best error is compared with the stopping criterion. If the results meet the stopping criterion, the methodology ends as shown in steps 10 and 11. If this is not the case, an improvement is made to the method that produced the worst result, in order to continue generating competitiveness with the methods, and depending on the method, the adjustment or improvement is made as shown in step 12, and step 13 begins a new cycle until finding the method that meets the stopping criterion.

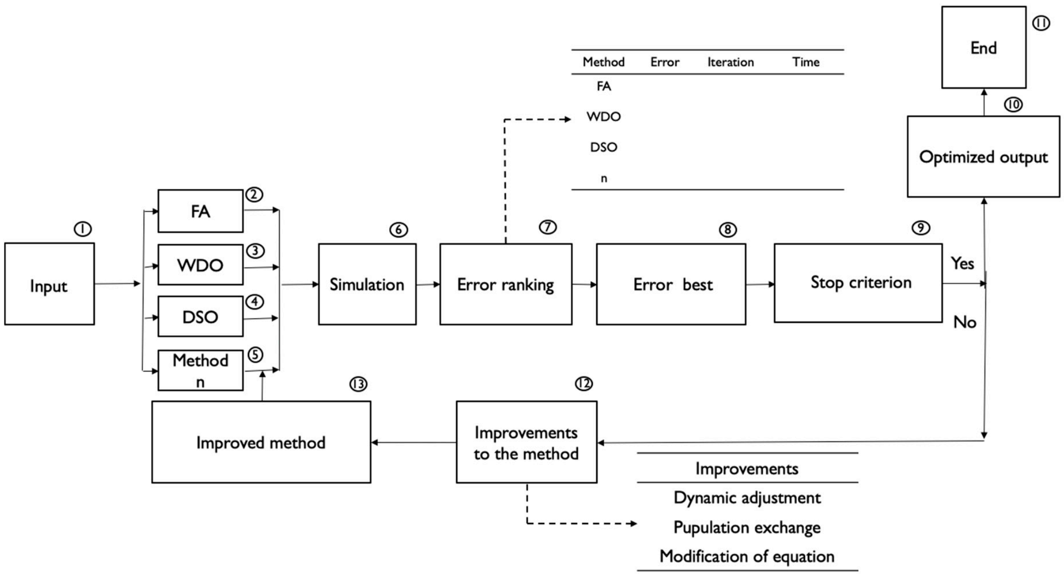

Figure 2. Data flow of the methodology.

3.1. Case 1: Benchmark Functions

In case 1, unimodal, multimodal, and fixed-dimension multimodal benchmark functions [41] were optimized, and their definitions are summarized in Table 2, Table 3, and Table 4, respectively. The functions are optimized using the aforementioned methods, and the competitiveness among them is compared, in this way finding out more detail about their operation and discovering the advantages and disadvantages of each method.

Table 2. Unimodal benchmark functions.

Benchmark functions $f1$–$f7$
$f_1(x) = \sum_{1=1}^{n} x_i^2$ *Search space* $x_j \in [-100,\ 100]$ *and* $f(x^*) = 0$
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ *Search space* $x_j \in [-10,\ 10]$ *and* $f(x^*) = 0$
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$ *Search space* $x_j \in [-100,\ 100] and f(x^*) = 0$
$f_4(x) = max_i\{\lvert x_i \rvert, ,\ 1 \le i \le n\}$ *Search space* $x_j \in [-100,\ 100]$ *and* $f(x^*) = 0$
$f_5(x) = \sum_{i=1}^{n-1} [100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2]$ *Search space* $x_j \in [-30,\ 30]$ *and* $f(x^*) = 0$
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$ *Search space* $x_j \in [-100,\ 100]$ *and* $f(x^*) = 0$
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1]$ *Search space* $x_j \in [-1.28,\ 1.28]$ *and* $f(x^*) = 0$

Table 3. Multimodal benchmark functions.

Benchmark functions *f9–f11*
$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$ *Search space* $x_j \in [-5.12,\ 5.12]$ *and* $f(x^*) = 0$
$f_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$ *Search space* $x_j \in [-32,\ 32]$ *and* $f(x^*) = 0$
$f_{11}(x) = \frac{1}{400}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ *Search space* $x_j \in [-600,\ 600]$ *and* $f(x^*) = 0$

Table 4. Fixed-dimension multimodal benchmark functions.

Benchmark functions *f15–f18*
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$ *Search space* $x_j \in [-5,\ 5]$ *and* $f(x^*) = 0.00030$
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$ *Search space* $x_j \in [-5,\ 5]$ *and* $f(x^*) = -1.0316$
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$ *Search space* $x_j \in [-5,\ 5]$ *and* $f(x^*) = 0.398$
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2] *$ $\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)]$ *Search space* $x_j \in [-2,\ 2]$ *and* $f(x^*) = 3$

3.2. Case 2 Fuzzy Controller

In this case, we propose the optimization for a fuzzy controller of an autonomous mobile robot, using the multi-metaheuristic competitiveness model to find out which is the satisfactory optimization algorithm for this specific problem; below, the fuzzy controller is explained in more detail.

This autonomous robot mobile controller is presented in [42,43]; in [44], some applications of the method are described. The controller aims at accurately tracking a given desired trajectory, and it has nine rules that allow it to create a relationship between the linguistic variables of the fuzzy system. The linguistic variables are as follows: in the first input, it is the linear velocity (ev); in the second and last input, it is the angular velocity (ew). The two inputs have the same linguistic values: positive (P), zero (Z), and negative (N), with the same membership functions in each one. The outputs are two: torque 1 (t1) and torque 2 (t2) represent the movement of the wheels when the robot rotates, and each of these outputs has three triangular functions.

Figure 3 shows the desired trajectory that the robot has to follow; this figure has the Y axis of the desired displacement line, and the X axis shows the displacement time.

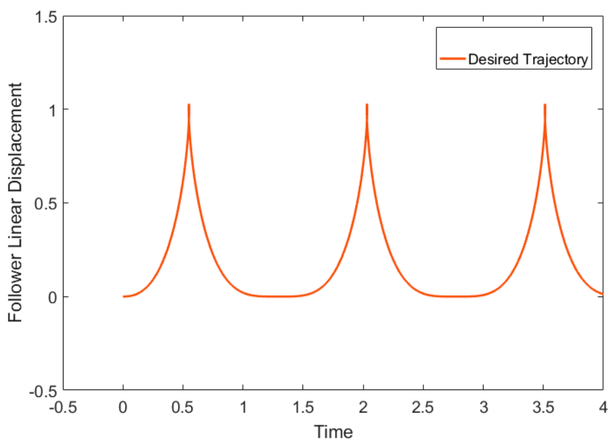

Figure 3. Desired trajectory in normalized axis.

Figure 4 shows the red line as the desired trajectory, and the actual trajectory of the robot in blue, and as can be seen, the robot is very much lost, since it generates a large error of $\mathbf{3.8541 \times 10^{+03}}$.

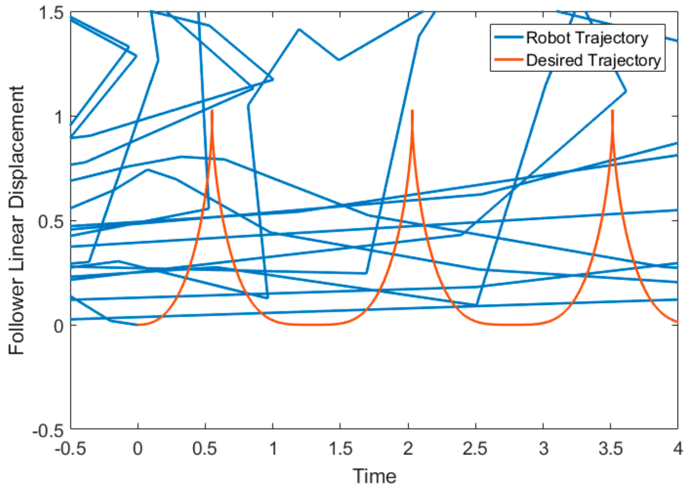

Figure 4. Obtained mean square error (MSE).

Figure 5 illustrates the membership functions of input 1, which is called the linear speed, where its linguistic variables represent negative, zero, and positive values.

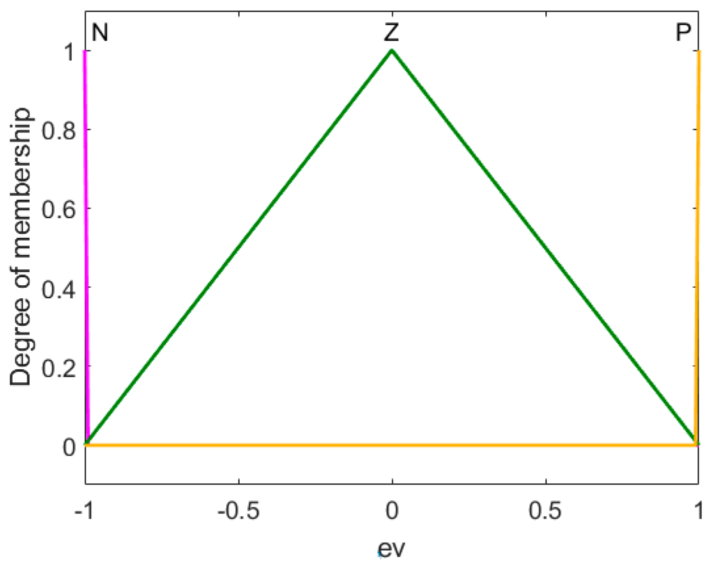

Figure 5. Input linear velocity (ev), not optimized.

Figure 6, we can find the membership functions; the blue color corresponds to a trapezoidal, the orange represents the zero linguistic variable and is a triangular membership function, and the yellow line is a trapezoidal membership function of input 2 of the fuzzy controller.

Figure 7 shows the three triangular membership functions of output 1, which represents the movement of wheel 1 of the fuzzy control, as illustrated by the functions that do not look well defined, because the controller is not optimized in this figure.

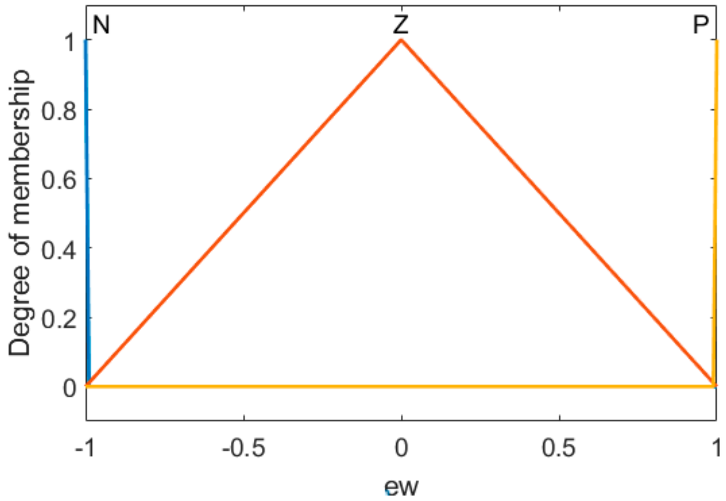

Figure 6. Input angular velocity (ew), not optimized.

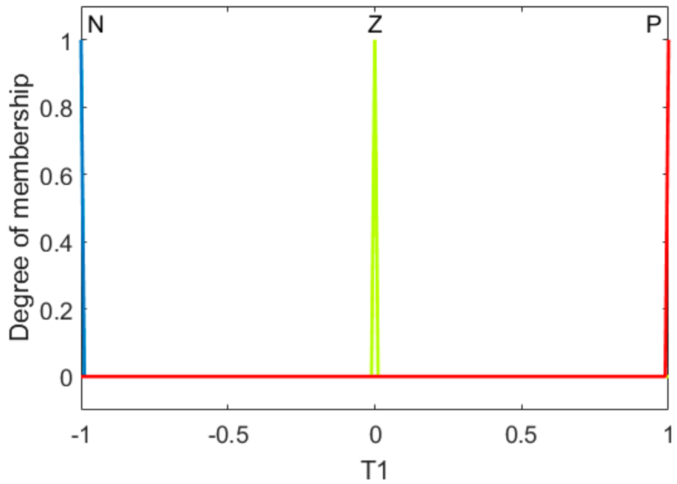

Figure 7. Output T1, not optimized.

Figure 8 represents that output 2 of the fuzzy controller (T2) represents the movement of wheel 2, and has negative, zero, and positive as linguistic variables. In this case, the three membership functions are triangular.

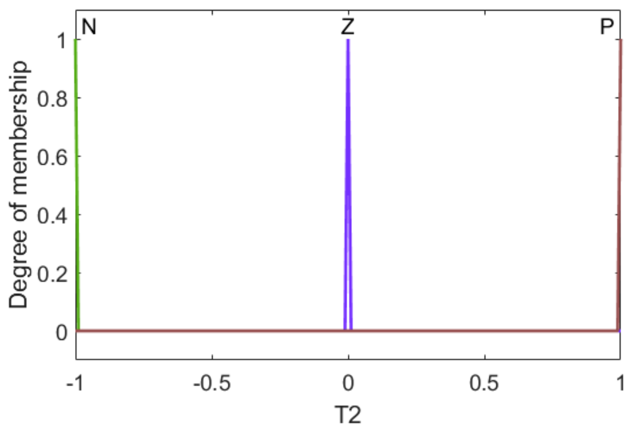

Figure 8. Output T2, not optimized.

Table 5 shows the representation of the if-then rules of the fuzzy controller that describe the relationship between the linguistic variables, where e_v represents the linear velocity and e_w is the angular velocity.

Table 5. If-then fuzzy rules.N: negative, P: positive, Z: zero.

If then Rules
1. If (ev is N) and (ew is N) then (T1 is N)(T2 is N) (1)
2. If (ev is N) and (ew is Z) then (T1 is N)(T2 is Z) (1)
3. If (ev is N) and (ew is P) then (T1 is N)(T2 is P) (1)
4. If (ev is Z) and (ew is N) then (T1 is Z)(T2 is N) (1)
5. If (ev is Z) and (ew is Z) then (T1 is Z)(T2 is Z) (1)
6. If (ev is Z) and (ew is P) then (T1 is Z)(T2 is P) (1)
7. If (ev is P) and (ew is N) then (T1 is P)(T2 is N) (1)
8. If (ev is P) and (ew is Z) then (T1 is P)(T2 is Z) (1)
9. If (ev is P) and (ew is P) then (T1 is P)(T2 is P) (1)

4. Results

This section is presented in two parts. The first is for case 1, which describes the results obtained by the optimization of benchmark functions using the three methods proposed for the methodology. The second part, for case 2, shows the results of the fuzzy controller optimization for the autonomous mobile robot. For this optimization, the method that optimizes generates a vector of data which goes to the parameters of the membership functions being optimized.

4.1. Case 1 Results: Benchmark Functions

Table 6 shows the parameters used for the benchmark functions. Column 1 of Table 7 shows the number that represents the particular function; column 2 shows the minimum of the function; columns 3 and 4 show the averages and standard deviations respectively obtained with the WDO; finally, columns 5 and 6 shows the results of DSO. Each of these functions was evaluated 30 times with the same parameters to obtain the averages and standard deviations. The parameters used in the WDO, DSO, and FA methods are presented in Table 8.

Table 6. Parameters used in the experiments.

Population	**Iterations**	**Dimensions**
30	500	30

Table 7. Results for 30 dimensions.

		WDO		**DSO**	
Function	**$f$min**	**Average**	**Standard Deviation**	**Average**	**Standard Deviation**
$f1$	0	3.66×10^{-27}	3.87×10^{-27}	4.95×10^{-09}	1.39×10^{-08}
$f2$	0	3.26×10^{-14}	2.70×10^{-14}	8.94×10^{-09}	2.68×10^{-08}
$f3$	0	2.42×10^{-20}	2.02×10^{-20}	1.16×10^{-03}	3.77×10^{-03}
$f4$	0	5.90×10^{-13}	6.18×10^{-13}	3.00×10^{-03}	6.76×10^{-03}
$f5$	0	28.5696	7.65×10^{-02}	$9.82 \times 10^{+00}$	$1.15 \times 10^{+01}$
$f6$	0	1.74×10^{-02}	4.52×10^{-03}	2.71×10^{-03}	2.14×10^{-05}
$f7$	0	1.12×10^{-02}	3.09×10^{-02}	2.71×10^{-03}	1.64×10^{-03}
$f9$	0	−118.27	0	$5.58 \times 10^{+00}$	$9.24 \times 10^{+00}$
$f10$	0	9.53×10^{-15}	1.23×10^{-14}	3.47×10^{-07}	7.22×10^{-07}
$f11$	0	0	0	3.66×10^{-11}	1.41×10^{-10}
$f15$	0.00030	3.07×10^{-04}	2.18×10^{-07}	3.07×10^{-04}	1.39×10^{-15}
$f16$	−1.0316	1.0316	6.78×10^{-16}	$-1.03 \times 10^{+00}$	4.59×10^{-16}
$f17$	0.398	0.3979	6.78×10^{-05}	0	
$f18$	3	7.7827	3.61×10^{-15}	3	3.26×10^{-15}

The WDO average and standard deviation results with 64 dimensions can be compared with the results obtained with 30 dimensions.

Table 8. Parameters used in the WDO, DSO, and FA methods.

Population	Iterations	Dimensions
30	500	64

Table 9. Simulations results for 64 dimensions.

		WDO		DSO	
Function	**$f$min**	**Average**	**Standard Deviation**	**Average**	**Standard Deviation**
$f1$	0	5.07×10^{-27}	8.47×10^{-27}	8.63×10^{-06}	2.16×10^{-05}
$f2$	0	4.57×10^{-14}	5.89×10^{-14}	1.50×10^{-03}	3.99×10^{-03}
$f3$	0	5.25×10^{-18}	5.66×10^{-18}	7.43×10^{-02}	2.04×10^{-01}
$f4$	0	2.32×10^{-12}	1.68×10^{-12}	7.80×10^{-03}	1.35×10^{-02}
$f5$	0	6.45×10^{-13}	5.42×10^{-13}	$1.54 \times 10^{+01}$	$2.34 \times 10^{+01}$
$f6$	0	7.34×10^{-02}	1.36×10^{-02}	1.59×10^{-05}	2.41×10^{-05}
$f7$	0	5.11×10^{-02}	9.11×10^{-03}	5.79×10^{-03}	3.78×10^{-03}
$f9$	0	$2.13 \times 10^{+02}$	$4.66 \times 10^{+01}$	$5.58 \times 10^{+01}$	$5.22 \times 10^{+01}$
$f10$	0	3.85×10^{-15}	6.05×10^{-15}	1.30×10^{-03}	2.41×10^{-03}
$f11$	0	$2.12 \times 10^{+01}$	2.70×10^{-01}	2.96×10^{-07}	1.05×10^{-06}
$f15$	0.00030	3.08×10^{-04}	1.68×10^{-07}	3.07×10^{-04}	2.37×10^{-16}
$f16$	−1.0316	$-1.03 \times 10^{+00}$	6.78×10^{-16}	$-1.03 \times 10^{+00}$	4.50×10^{-16}
$f17$	0.398	3.98×10^{-01}	4.90×10^{-06}	3.98×10^{-01}	0
$f18$	3	7.7827	3.61×10^{-15}	3	6.54×10^{-16}

Table 9 above shows the optimization results of benchmark functions where the minimum of functions f15, f16, f17, and f18 is different from zero. With the evaluation of functions with different minima, the performance of the optimization in the functions with the WDO and DSO can be better observed.

Table 10 shows the parameters used in the above-mentioned methods. These parameters are the same for each of these algorithms as a fair form of competition is aimed.

Table 10. Population, iterations, and dimensions.

Population	Iterations	Dimensions
30	500	128

Table 11 shows the results of 30 experiments, where the performance of the WDO and DSO method is compared, using 128 dimensions for the search evaluation of the objective function, which is in this case the minimum of each function; these are shown in column 1 of the table.

The following Tables 12 and 13 show the results obtained with the FA for the optimization of the f1 and f2 function with 30, 64, and 128 dimensions, respectively, where it can be noted that the results are very far from the global minima of the functions. Previously, we have been experimenting with this FA method, where we can note that in order to obtain a good result with it, it is necessary to increase the iterations and maintain a population of 30 to 50 fireflies. Therefore, as future work in this methodology, an adjustment will be made to the method to improve its behavior in the optimization of benchmark functions, remembering that this methodology is proposed just for that: to find which method is good for a specific problem and in competitiveness with others. For the method with which a worse result is obtained, we can make an improvement to help its behavior.

Table 11. Results for 128 dimensions.

		WDO		DSO	
Function	f**min**	**Average**	**Standard Deviation**	**Average**	**Standard Deviation**
$f1$	0	4.35×10^{-27}	7.16×10^{-27}	1.49×10^{-04}	7.68×10^{-04}
$f2$	0	3.88×10^{-14}	4.38×10^{-14}	2.66×10^{-02}	4.83×10^{-02}
$f3$	0	1.09×10^{-16}	1.07×10^{-16}	$1.21 \times 10^{+02}$	$3.34 \times 10^{+02}$
$f4$	0	1.52×10^{-12}	1.62×10^{-12}	2.56×10^{-02}	5.11×10^{-02}
$f5$	0	7.68×10^{-13}	4.61×10^{-13}	$4.60 \times 10^{+01}$	$5.62 \times 10^{+01}$
$f6$	0	1.85×10^{-01}	3.26×10^{-02}	7.43×10^{-01}	7.34×10^{-01}
$f7$	0	1.66×10^{-01}	3.11×10^{-02}	6.83×10^{-03}	6.49×10^{-03}
$f9$	0	$4.58 \times 10^{+02}$	$2.09 \times 10^{+02}$	$8.38 \times 10^{+01}$	$1.19 \times 10^{+02}$
$f10$	0	8.88×10^{-16}	4.01×10^{-16}	2.12×10^{-04}	7.98×10^{-04}
$f11$	0	$2.07 \times 10^{+01}$	$3.91 \times 10^{+00}$	1.48×10^{-03}	3.93×10^{-03}
$f15$	0.00030	3.08×10^{-04}	1.93×10^{-07}	3.07×10^{-04}	1.39×10^{-16}
$f16$	−1.0316	$-1.03 \times 10^{+00}$	8.78×10^{-16}	$-1.03 \times 10^{+00}$	4.70×10^{-16}
$f17$	0.398	3.98×10^{-01}	3.46×10^{-06}	3.98×10^{-01}	$0.00 \times 10^{+00}$
$f18$	3	$7.78 \times 10^{+00}$	3.61×10^{-15}	$3.00 \times 10^{+00}$	4.79×10^{-06}

Table 12. Results for f1 with FA.

Figure	**f1**	**f1**	**f1**
Iterations	500	500	500
Dimensions	30	64	128
Average	8.80×10^{-03}	8.40×10^{-03}	3.67×10^{-01}
Standard Deviation	1.90×10^{-03}	1.20×10^{-02}	2.93×10^{-02}

Table 13. Results for f2 with FA.

Function	f2	f2	f2
Iterations	500	500	500
Dimensions	30	64	128
Average	3.20×10^{-01}	1.84×10^{00}	1.45×10^{00}
Standard Deviation	6.92×10^{-02}	2.58×10^{-01}	2.00×10^{-01}

4.2. Case 2 Results of the Fuzzy Controller Optimization

Table 14 shows the parameters used in the FA, WDO, and DSO methods, and Table 15 shows the 30 experiments performed to obtain the best optimized fuzzy system, using as metric the MSE, where it can be observed that the best error found is of 0.00169, and in general, the values only varied from 10^{-02} to 10^{-03}.

Table 14. Parameters used in FA, WDO, and DSO for fuzzy controller optimization.

Population	**Iterations**
20	1500

In Figures 9 and 10, the variation that was made in the parameters of the membership functions of the two inputs of the fuzzy controller is observed, where the uncertainty that exists between each of them can be observed, and with this, the error that generates the fuzzy controller in the simulation is considerably improved. Figures 11 and 12 show the outputs of the controller optimized by the FA, respectively.

Table 15. Results of the experiments with the fuzzy controller.

Experiment	MSE	Experiment	MSE
1	1.96×10^{-03}	16	2.99×10^{-02}
2	4.80×10^{-03}	17	3.33×10^{-02}
3	6.31×10^{-03}	18	4.09×10^{-02}
4	6.33×10^{-03}	19	4.32×10^{-02}
5	8.25×10^{-03}	20	5.33×10^{-02}
6	9.09×10^{-03}	21	7.11×10^{-02}
7	9.58×10^{-03}	22	7.06×10^{-02}
8	1.71×10^{-02}	23	7.72×10^{-02}
9	1.82×10^{-02}	24	8.03×10^{-02}
10	1.85×10^{-02}	25	8.47×10^{-02}
11	1.98×10^{-02}	26	8.75×10^{-02}
12	2.11×10^{-02}	27	8.97×10^{-02}
13	2.27×10^{-02}	28	9.16×10^{-01}
14	2.45×10^{-02}	29	9.42×10^{-02}
15	2.52×10^{-02}	30	9.63×10^{-02}

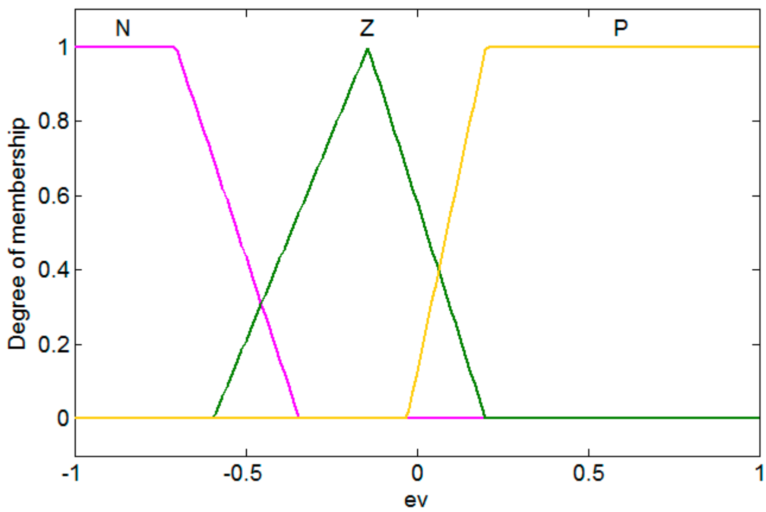

Figure 9. Input 1 (ev).

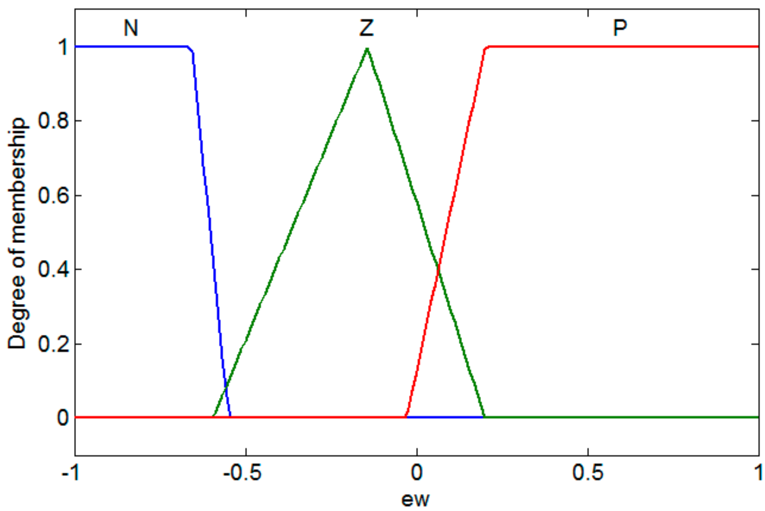

Figure 10. Input 2 (ew).

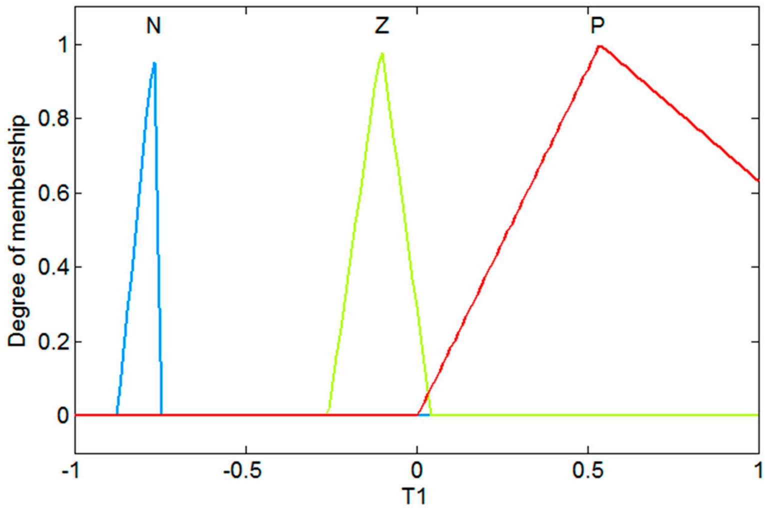

Figure 11. Output 1 (T1).

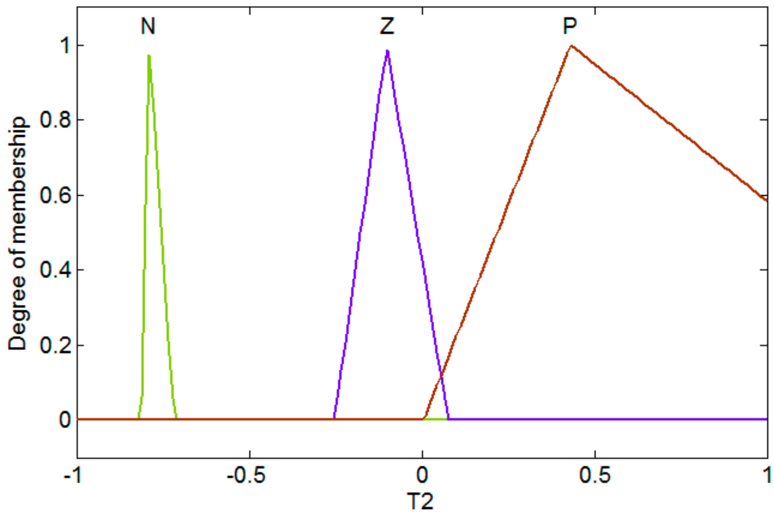

Figure 12. Output 2 (T2).

As can be noted in Table 16, the results obtained with the WDO method were very far from each other, since it has the best MSE of 0.000019, but as the worst is 0.057094. For this reason, on average, this method cannot be the best at optimizing this specific problem.

Table 16. Results obtained with WDO.

Experiment	MSE	Experiment	MSE
1	1.90×10^{-05}	16	1.64×10^{-04}
2	1.90×10^{-05}	17	1.64×10^{-04}
3	5.40×10^{-05}	18	2.60×10^{-04}
4	5.40×10^{-05}	19	2.93×10^{-04}
5	6.20×10^{-05}	20	4.33×10^{-04}
6	6.20×10^{-05}	21	1.62×10^{-03}
7	6.90×10^{-05}	22	1.65×10^{-03}
8	8.70×10^{-05}	23	6.21×10^{-03}
9	8.70×10^{-05}	24	6.35×10^{-03}
10	9.10×10^{-05}	25	7.73×10^{-03}
11	9.80×10^{-05}	26	1.67×10^{-02}
12	1.05×10^{-04}	27	1.67×10^{-02}
13	1.15×10^{-04}	28	2.65×10^{-02}
14	1.38×10^{-04}	29	2.92×10^{-02}
15	1.60×10^{-04}	30	5.70×10^{-02}

Figures 13–16 show the optimized movement of the parameters of the membership functions of the inputs e_v, e_w and the outputs t1 and t2, respectively.

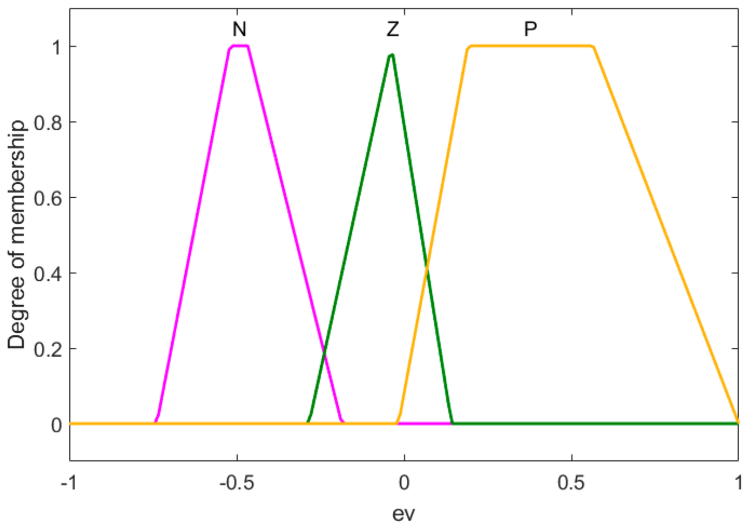

Figure 13. Membership functions for Input 1.

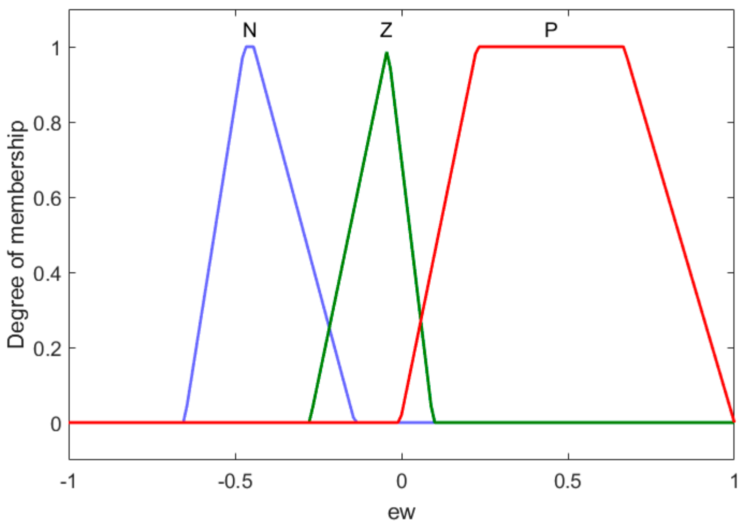

Figure 14. Membership functions for Input 2.

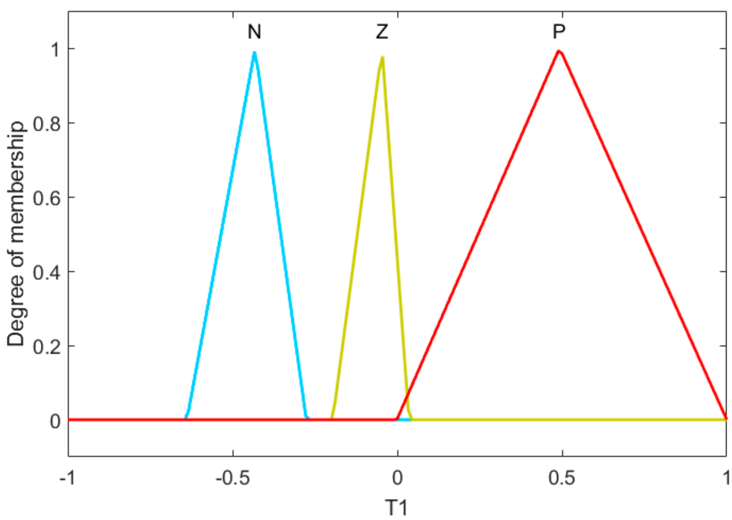

Figure 15. Membership functions for Output 1.

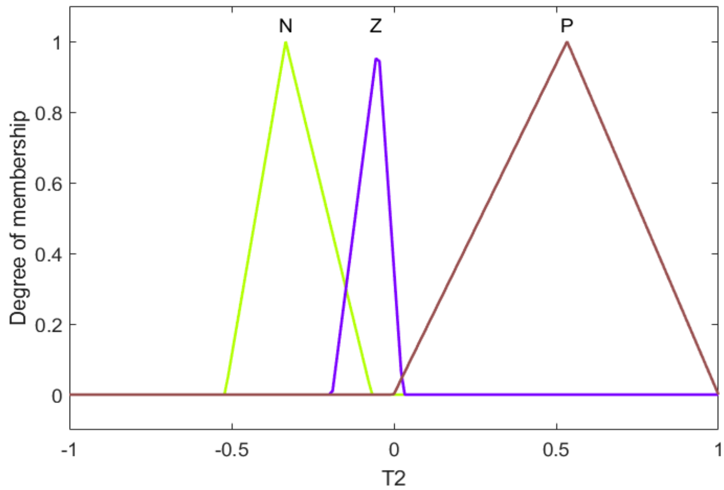

Figure 16. Membership functions for Output 2.

DSO on average gives a better result than the FA and WDO. As can be noted in column 2 in Table 17, it appears that this is not the case, since the WDO method gives much lower values, but they are very separated from each other, and the DSO remains consistent with the delivered results of the MSE.

Table 17. Results obtained with DSO.

Experiment	MSE	Experiment	MSE
1	1.69×10^{-03}	16	2.99×10^{-02}
2	4.80×10^{-03}	17	3.33×10^{-02}
3	6.31×10^{-03}	18	4.09×10^{-02}
4	6.33×10^{-03}	19	4.32×10^{-02}
5	8.25×10^{-03}	20	5.33×10^{-02}
6	9.09×10^{-03}	21	7.11×10^{-02}
7	9.58×10^{-03}	22	7.60×10^{-02}
8	1.71×10^{-02}	23	7.72×10^{-02}
9	1.82×10^{-02}	24	8.03×10^{-02}
10	1.85×10^{-02}	25	8.04×12^{-02}
11	1.98×10^{-02}	26	8.75×10^{-02}
12	2.11×10^{-02}	27	8.97×10^{-02}
13	2.27×10^{-02}	28	9.16×10^{-02}
14	2.45×10^{-02}	29	9.42×10^{-02}
15	2.52×10^{-02}	30	9.63×10^{-02}

As can be noted from Figures 17–20, the overlap between existing functions helps the robot to have a better tracking of the desired trajectory, in comparison with the other methods used in this methodology.

Table 18 summarized the results of the optimization methods.

Table 18. Summary of results from the optimization methods.

	FA	WDO	DSO
Average	4.21×10^{-02}	5.78×10^{-03}	5.02×10^{-02}
Standard deviation	3.31×10^{-02}	1.24×10^{-02}	3.3×10^{-02}

The equation for the Z test is as follows:

$$Z = \frac{(\overline{x}_1 - \overline{x}_2) - (\mu_1 - \mu_2)}{\sigma_{\overline{x}_1 - \overline{x}_2}} \tag{9}$$

Table 19 shows the statistical data used in the z test, and Table 20 the results of the test.

Table 19. Statistical data.

Variable	Number of Samples	Mean	Standard Deviation
WDO	30	0.006	0.012
DSO	30	0.042	0.033

Table 20. Parameters statistical test.

Parameters	Value
Difference	−0.036
z (Observed value)	−5.628
z (Critica value)	−1.645
valor-p (one-tailed)	<0.0001
alpha	0.05

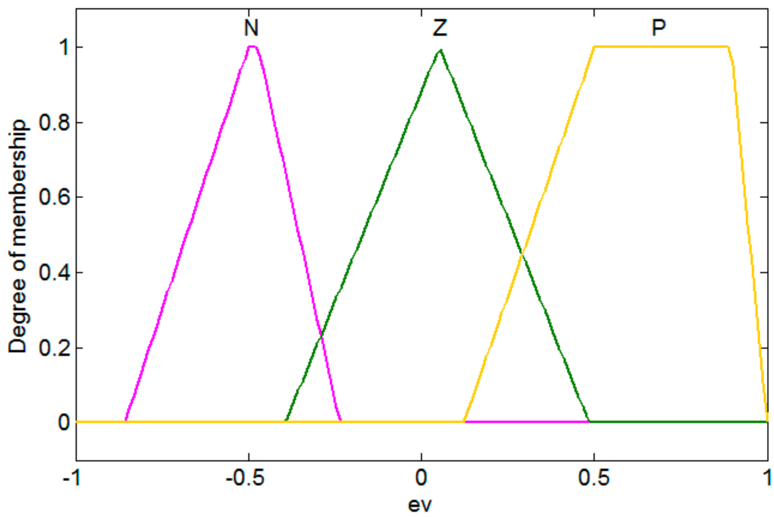

Figure 17. Input 1 optimized by DSO.

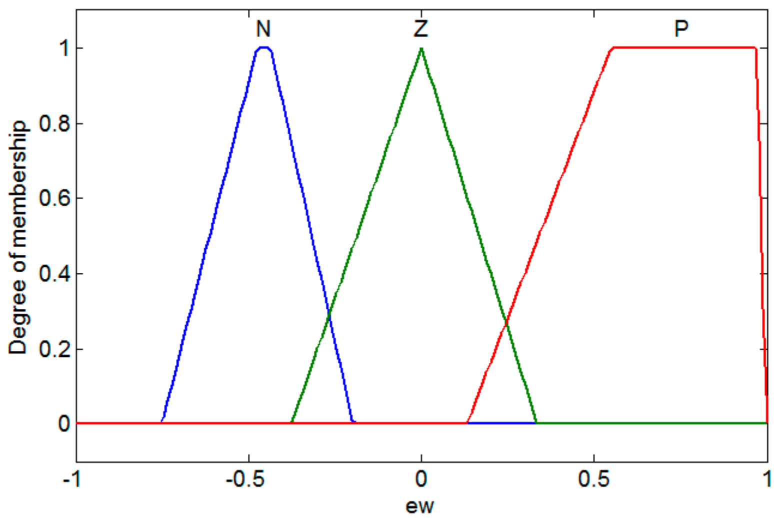

Figure 18. Input 2 optimized by DSO.

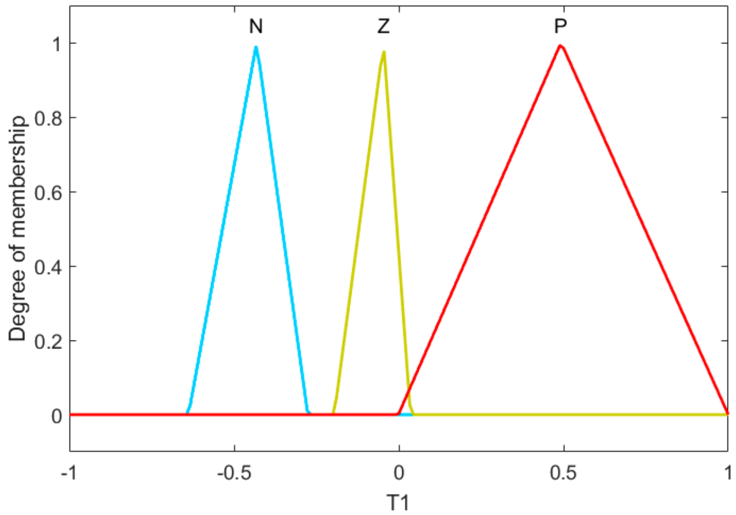

Figure 19. Output 1 optimized by DSO.

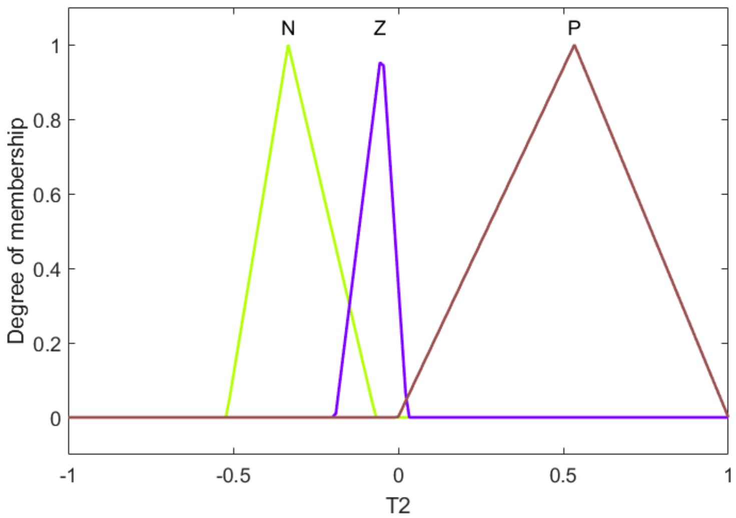

Figure 20. Output 1 optimized by DSO.

5. Discussion

The methodology proposes competitiveness among optimization metaheuristics to improve overall performance, and this is aimed at reducing the search time from one method to another when one of them does not give the expected results. In addition to the search time, the superior method is also obtained with more certainty for each type of problem optimization, since the competitiveness that exists between them delivers only the best result. With this methodology, the method that generates the worst result is also helped, also assisting other scientists regarding the cases for which the used methods are good and for which they are not. Therefore, with the obtained results in the optimization of unimodal, multimodal, and fixed-dimension multimodal benchmark functions, it can be said that the WDO in general gives, on average, better results for this type of function, while the DSO algorithm gives more results that are far from the minimum of the function. However, in terms of the parameter optimization of the membership functions of a fuzzy controller in particular, as can noted in Table 18, the method that on average produces better results is DSO. The method that can be said to be the worst regarding these results of competitiveness is the FA, since this method did not show good results with any of the two previous cases of minimization. For this reason, as future work, it is proposed to make an improvement to this method, depending on the previous research that has its disadvantages in these cases.

6. Conclusions

The proposed methodology was created to optimize problems, and was tested with the unimodal and multimodal benchmark functions with the WDO, DSO, and FA methods. Each of them was put into competition with the same parameters for a fair competition, with 30 experiments as the limitation. Under these parameters, it was obtained that WSO was better for benchmark functions. On the other hand, for the optimization of the parameters of the membership functions, the DSO method was better, and it was the metaheuristic that found the data vector that managed to optimize the functions of the fuzzy controller in such a way that the robot in simulation approached the desired trajectory. The proposed methodology showed which method in the competition was the best to solve a specific case, and was expected to improve the method with which good results were not obtained. As future work, we plan to perform more experimentation with other optimization problems. In addition, it is worth mentioning that other methods for the required optimization can be added in this methodology. The proposed methodology for optimization problems is the main contribution of this work, resulting in the best method among the competition.

Author Contributions: F.V. and J.S. reviewed the state of the art; O.C. contributed to the discussion and analysis of the results; M.L.L. analyzed of the original method and used fuzzy logic for parameter adaptation, contributed to the simulations and wrote the paper. All authors have read and approved the final manuscript.

Funding: This research was funded by CONACYT under grant 122.

Acknowledgments: We would like to express our gratitude to CONACYT, Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

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

References

1. Lukovac, V.; Pamučar, D.; Popović, M.; Đorović, B. Portfolio model for analyzing human resources: An approach based on neuro-fuzzy modeling and the simulated annealing algorithm. *Expert Syst. Appl.* **2017**, *90*, 318–331. [CrossRef]
2. Pamučar, D.; Vasin, L.; Atanasković, P.; Miličić, M. Planning the City Logistics Terminal Location by Applying the Green p-Median Model and Type-2 Neurofuzzy Network. *Comput. Intell. Neurosci.* **2016**, *2016*, 1–15. [CrossRef]
3. Sremac, S.; Tanackov, I.; Kopić, M.; Radović, D. ANFIS model for determining the economic order quantity. *Decis. Mak. Appl. Manag. Eng.* **2018**, *2*, 81–92. [CrossRef]
4. Stojčić, M.; Stjepanović, A.; Stjepanović, Đ. ANFIS model for the prediction of generated electricity of photovoltaic modules. *Decis. Mak. Appl. Manag. Eng.* **2019**, *1*, 35–48. [CrossRef]
5. Pamucar, D.; Ćirović, G. Vehicle route selection with an adaptive neuro fuzzy inference system in uncertainty conditions. *Decis. Mak. Appl. Manag. Eng.* **2018**, *1*, 13–37. [CrossRef]
6. Pamučar, D.; Ljubojević, S.; Kostadinović, D.; Đorović, B. Cost and risk aggregation in multi-objective route planning for hazardous materials transportation—A neuro-fuzzy and artificial bee colony approach. *Expert Syst. Appl.* **2016**, *65*, 1–15. [CrossRef]
7. Pamučar, D.; Atanasković, P.; Miličić, M. Modeling of fuzzy logic system for investment management in the railway infrastructure. *Teh. Vjesn. Tech. Gaz.* **2015**, *22*. [CrossRef]
8. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [CrossRef]
9. Liu, J.; Yang, J.; Liu, H.; Tian, X.; Gao, M. An improved ant colony algorithm for robot path planning. *Soft Comput.* **2017**, *21*, 5829–5839. [CrossRef]
10. Dorigo, M.; Birattari, M.; Blum, C.; Clerc, M.; Stutzle, T.; Winfield, A. Ant colony optimization and swarm intelligence. In Proceedings of the ANTS 2008: The 6th International Conference on Ant Colony Optimization and Swarm Intelligence, Brussels, Belgium, 22–24 September 2008; Springer: Berlin, Germany, 2008; Volume 5217.
11. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.

12. Pal, S.K.; Wang, P.P. *Genetic Algorithms for Pattern Recognition*; CRC Press: Boca Raton, FL, USA, 2017.
13. Garg, H. A hybrid GSA-GA algorithm for constrained optimization problems. *Inf. Sci.* **2019**, *478*, 499–523. [CrossRef]
14. Patwal, R.S.; Narang, N.; Garg, H. A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy* **2018**, *142*, 822–837. [CrossRef]
15. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [CrossRef]
16. Garg, H.; Sharma, S.P. Multi-objective reliability-redundancy allocation problem using particle swarm optimization. *Comput. Ind. Eng.* **2013**, *61*, 247–255. [CrossRef]
17. Garg, H. Solving structural engineering design optimization problems using an artificial bee colony algorithm. *J. Ind. Manag. Optim.* **2014**, *3*, 777–794. [CrossRef]
18. Lagunes, M.L.; Castillo, O.; Soria, J.; Garcia, M.; Valdez, F. Optimization of granulation for fuzzy controllers of autonomous mobile robots using the Firefly Algorithm. *Granul. Comput.* **2019**, *4*, 185–195. [CrossRef]
19. Bernal, E.; Castillo, O.; Soria, J. Imperialist Competitive Algorithm with Dynamic Parameter Adaptation Applied to the Optimization of Mathematical Functions. In *Nature-Inspired Design of Hybrid Intelligent Systems*; Springer: Cham, Switzerland, 2017; Volume 667, pp. 329–341.
20. Caraveo, C.; Valdez, F.; Castillo, O. A New Meta-Heuristics of Optimization with Dynamic Adaptation of Parameters Using Type-2 Fuzzy Logic for Trajectory Control of a Mobile Robot. *Algorithms* **2017**, *10*, 85. [CrossRef]
21. Olivas, F.; Amador-Angulo, L.; Perez, J.; Careveo, C.; Valdez, F.; Castillo, O. Comparative Study of Type-2 Fuzzy Particle Swarm, Bee Colony and Bat Algorithms in Optimization of Fuzzy Controllers. *Algorithms* **2017**, *10*, 101. [CrossRef]
22. Lagunes, M.L.; Castillo, O.; Soria, J. Methodology for the Optimization of a Fuzzy Controller Using a Bio-inspired Algorithm. In Proceedings of the North American Fuzzy Information Processing Society Annual Conference, Cancun, Mexico, 16 October 2017; Springer: Cham, Switzerland, 2018; Volume 648, pp. 131–137.
23. Yang, X.-S. Firefly Algorithm, Lévy Flights and Global Optimization. In *Research and Development in Intelligent Systems XXVI*; Springer: London, UK, 2010; pp. 209–218.
24. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 1308–3898. [CrossRef]
25. Gandomi, A.H.; Yang, X.-S.; Alavi, A.H. Mixed variable structural optimization using Firefly Algorithm. *Comput. Struct.* **2011**, *89*, 2325–2336. [CrossRef]
26. Bayraktar, Z.; Komurcu, M.; Bossard, J.A.; Werner, D.H. The Wind Driven Optimization Technique and its Application in Electromagnetics. *IEEE Trans. Antennas Propag.* **2013**, *61*, 2745–2757. [CrossRef]
27. Bayraktar, Z.; Komurcu, M.; Werner, D.H. Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics. In Proceedings of the 2010 IEEE Antennas and Propagation Society International Symposium, Toronto, ON, Canada, 11–17 July 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–4.
28. De Melo, V.V.; Banzhaf, W. Drone Squadron Optimization: A novel self-adaptive algorithm for global numerical optimization. *Neural Comput. Appl.* **2018**, *30*, 3117–3144. [CrossRef]
29. De Melo, V.V. A novel metaheuristic method for solving constrained engineering optimization problems: Drone Squadron Optimization. *arXiv* **2017**, arXiv:1708.01368.
30. Yalcin, Y.; Pekcan, O. Nuclear Fission-Nuclear Fusion algorithm for global optimization: A modified Big Bang-Big Crunch algorithm. *Neural Comput. Appl.* **2018**, *31*, 1–33. [CrossRef]
31. Shi, Y. Particle swarm optimization: Developments, applications and resources. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Korea, 27–30 May 2001; IEEE: Piscataway, NJ, USA, 2001; Volume 1, pp. 81–86.
32. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory, in MHS'95. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 39–43.
33. Gong, M.; Cai, Q.; Chen, X.; Ma, L. Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition. *IEEE Trans. Evol. Comput.* **2014**, *18*, 82–97. [CrossRef]
34. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]

35. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning-III. *Inf. Sci.* **1975**, *9*, 43–80. [CrossRef]
36. Zadeh, L.A. Fuzzy logic. *Computer* **1988**, *21*, 83–93. [CrossRef]
37. Ibrahim, M.T.; Hanafi, D.; Ghoni, R. Autonomous Navigation for a Dynamical Hexapod Robot Using Fuzzy Logic Controller. *Procedia Eng.* **2012**, *38*, 330–341. [CrossRef]
38. Garcia, M.P.; Montiel, O.; Castillo, O.; Sepúlveda, R.; Melin, P. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl. Soft Comput.* **2009**, *9*, 1102–1110. [CrossRef]
39. Al-Jarrah, R.; Shahzad, A.; Roth, H. Path Planning and Motion Coordination for Multi-Robots System Using Probabilistic Neuro-Fuzzy. *IFAC-PapersOnLine* **2015**, *48*, 46–51. [CrossRef]
40. El Ferik, S.; Nasir, M.T.; Baroudi, U. A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space. *Appl. Soft Comput.* **2016**, *44*, 117–127. [CrossRef]
41. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
42. Astudillo, L.; Melin, P.; Castillo, O. Optimization of a Fuzzy Tracking Controller for an Autonomous Mobile Robot under Perturbed Torques by Means of a Chemical Optimization Paradigm. In *Recent Advances on Hybrid Intelligent Systems*; Springer: Heidelberg/Berlin, Germany, 2013; pp. 3–20.
43. Lagunes, M.L.; Castillo, O.; Soria, J. Optimization of Membership Function Parameters for Fuzzy Controllers of an Autonomous Mobile Robot Using the Firefly Algorithm. In *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*; Springer: Berlin, Germany, 2018; pp. 199–206.
44. 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 Proceedings of the North American Fuzzy Information Processing Society Annual Conference, Cancun, Mexico, 16 October 2017; Springer: Berlin, Germany; IEEE: Piscataway, NJ, USA, 2018; pp. 569–579.