





Article

Shadowed Type-2 Fuzzy Systems for Dynamic Parameter Adaptation in Harmony Search and Differential Evolution for Optimal Design of Fuzzy Controllers

Oscar Castillo ¹, Cinthia Peraza ¹, Patricia Ochoa ¹, Leticia Amador-Angulo ¹, Patricia Melin ¹, Yongjin Park ² and Zong Woo Geem ^{3,*}

¹ Division of Graduate Studies and Research, Tijuana Institute of Technology, Tijuana 22379, Mexico; ocastillo@tectijuana.mx (O.C.); cinthia.peraza@tectijuana.edu.mx (C.P.); martha.ochoa18@tectijuana.edu.mx (P.O.); gloria.amador@tectijuana.edu.mx (L.A.-A.); pmelin@tectijuana.mx (P.M.)

² Department of Transportation Engineering, Keimyung University, Daegu 42601, Korea; ypark@kmu.ac.kr

³ College of IT Convergence, Gachon University, Seongnam 13120, Korea

* Correspondence: geem@gachon.ac.kr

Abstract: This article mainly focuses on the utilization of shadowed type-2 fuzzy systems used to achieve the goal of dynamically adapting the parameters of two already known algorithms in the literature: the harmony search and the differential evolution algorithms. It has already been established that type-2 fuzzy logic enhances the performance of metaheuristics by enabling parameter adaptation; however, the utilization of fuzzy logic results in an increased execution time. For this reason, in this article, the shadowed type-2 fuzzy approach is put forward as a way of reducing execution time, while maintaining the good results that the complete type-2 fuzzy model produces. The harmony search and differential evolution algorithms with shadowed type-2 parameter adaptations were applied to the problem of optimally designing fuzzy controllers. The simulations were performed with the controllers working in an ideal situation, and then with a real situation under different noise levels in order to reach a conclusion regarding the performance of each of the algorithms that were applied.

Keywords: shadowed type-2 fuzzy systems; harmony search; differential evolution algorithm



Citation: Castillo, O.; Peraza, C.; Ochoa, P.; Amador-Angulo, L.; Melin, P.; Park, Y.; Geem, Z.W. Shadowed Type-2 Fuzzy Systems for Dynamic Parameter Adaptation in Harmony Search and Differential Evolution for Optimal Design of Fuzzy Controllers. *Mathematics* **2021**, *9*, 2439. <https://doi.org/10.3390/math9192439>

Academic Editor: Basil Papadopoulos

Received: 26 August 2021

Accepted: 21 September 2021

Published: 1 October 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/>).

1. Introduction

Nowadays, is becoming more common to find new algorithms that are developed based on the behavior of nature, which can offer computational solutions to complex optimization problems. In this work, two algorithms based on two different approaches are considered: the harmony search (HS) and the differential evolution (DE) algorithms, which have already been compared in previous studies using different types of fuzzy systems and for different problems, both for control and mathematical functions. Some of the studies that have considered these two algorithms together are the following: the first discussed the optimal design of fuzzy systems using differential evolution and harmony search algorithms with dynamic parameter adaptations [1]. The next study considered shadowed type-2 fuzzy systems for dynamic parameter adaptation in the harmony search and differential evolution algorithms [2]. Additionally, a comparative study demonstrating fuzzy controller optimization using the bee colony, differential evolution, and harmony search algorithms was presented in [3], and a high-speed interval type-2 fuzzy system approach for dynamic parameter adaptation in metaheuristics was outlined in [4].

The above-mentioned works all use fuzzy logic, which was first proposed in 1965 by Lotfi Zadeh and was immediately shown to be an important tool for the control of industrial systems and processes. In addition to the previously mentioned studies, fuzzy logic can also be applied in other areas to obtain better results, such as in industry, medicine,

and system control, just to mention a few. These are some of the works in which the use of fuzzy logic can be appreciated, but the evolution of fuzzy logic is very broad, and today, there is recent research in which dynamic fractional-order models are obtained based on an extension of general type-2 fuzzy systems, which are called interval type-3 fuzzy logic systems (IT3FLSs) [5]. These are also some of the most recent works in the literature in which the use of fuzzy logic has been put forward as an option for optimizing traditional problems, for example: an efficient type-2 fuzzy method for parameter adaptation in HS for optimally designing fuzzy controllers is presented in [6], an optimization of type-2 fuzzy controllers utilizing the galactic swarm and firefly algorithms is described in [7], the optimal parameter estimation of a silicon solar cell using fuzzy systems is described in [8], the application of soft computing to calculate medicine dose during patient treatment is shown in [9], a mathematical approach to identify coronavirus disease (COVID-19) using a fuzzy logic inference system is shown in [10], a fuzzy approach for preanesthetic operation risk evaluation in laparoscopic cholecystectomy is demonstrated in [11], fuzzy logic for estimating medicine imports can be seen in [12], the optimization of storage costs in maintenance for industry is presented in [13], an approach for job scheduling with a multi-criteria-based method is described in [14], an approach for supply chain management utilizing a fuzzy rule-based maturity model is shown in [15], the determination of the difficulty level for a garment model with a fuzzy logic method is presented in [16], and simulated annealing to locate electric charging situations with the implementation of interval type-2 fuzzy sets is presented in [17].

However, each of the metaheuristic algorithms that are used in this work have also had a great impact and have evolved separately in different areas of opportunity. First, we mention some recent research studies where the harmony search algorithm has been used: a differential HS algorithm for multi-objective optimal power flow is presented in [18], multilevel thresholding for image segmentation using the HS algorithm is described in [19], a hybrid HS algorithm for social network contact for COVID-19 tracing is shown in [20], a review of the HS algorithm in clustering problems is shown in [21], a hybrid of the HS algorithm and evolutionary approach for the process of feature selection is presented in [22], a neural-network-adaptive dynamical HS algorithm to approximate the flying rocks resulting from blasting is shown in [23], a selective harmonic elimination in output voltage levels in modular converters using a differential HS algorithm is shown in [24], a quasi-oppositional HS algorithm for ad hoc and sensor networks is presented in [25], a self-adaptive global-best HS algorithm-based airflow control is described in [26], a multi-population HS approach for the detection of high-order interactions is described in [27], the selection of features for the detection of colon cancer using clustering and a modification of the HS algorithm is shown in [28], feature selection with improved HS binary for data classification is described in [29], a multi-objective HS algorithm for the optimal expansion and planning of distribution systems is presented in [30], a hybrid HS algorithm for achieving an efficient healthcare framework for kidney disease is described in [31], an optimal steel frame design using an HS algorithm is described in [32], a discrete HS algorithm for a flexible job shop scheduling problem with multiple objectives is presented in [33], and a similarity hybrid harmony search algorithm for the Team Orienteering problem is presented in [34].

Below, we mention some works in which the DE algorithm has been used: a DE algorithm with wavelet functions and a novel mutation approach for optimization is presented in [35], an enhanced DE algorithm and its application in an optimization problem is shown in [36], the prediction of the velocity of a nanofluid convective flow by the DE algorithm in conjunction with a fuzzy system is presented in [37], a hybrid slime mold algorithm with a DE algorithm for optimization is described in [38], the optimization of operating rules in multi-reservoir hydropower generation with an adaptive DE algorithm is described in [39], the optimization of quantum cloning circuit parameters with an adaptive DE algorithm is shown in [40], an enhanced quantum-inspired DE algorithm for deep belief networks is described in [41], an enhanced multi-operator DE algorithm for solving

unconstrained problems is shown in [42], the optimal design of fuzzy controllers based on a DE algorithm with parameter adaptation utilizing fuzzy systems (of the type-2 and type-1 form) is presented in [43], and the accelerated DE algorithm with novel proposed operators for multi-damage detection in plate structures is described in [44].

It has already been established in previous works that type-2 fuzzy enhancements can help augment performance in metaheuristics by enabling dynamic parameter adaptation during execution. However, the utilization of fuzzy logic in parameter adaptation results in an increased execution time for the metaheuristic. For this reason, in this article, the shadowed type-2 fuzzy approach is proposed as a way of reducing execution time, while maintaining the good results of the complete type-2 fuzzy model in parameter adaptation. In summary, the most important contribution of this paper is the shadowed fuzzy approach proposed as a way to efficiently achieve parameter adaptation in HS and DE, and also the comparison of their performance in optimizing fuzzy controllers. Thus, in this paper, the objective of the work is the application of shadowed type-2 fuzzy systems as an approximation method of the type-2 general fuzzy systems modeled with the representation of α planes applied to the adjustment of parameters of the HS and DE algorithm. The main difference with respect to existing works is that the shadowed type-2 fuzzy inference system approach only requires two α -planes to model the general type-2 fuzzy inference systems, and another important contribution of the use of shadowed type-2 fuzzy systems is that the computational cost is much lower, and therefore, it is very feasible that they can be applied in real-life problems.

The rest of the article is organized as follows. Section 2 outlines the basic constructs of type-2 shadowed sets theory, Section 3 briefly describes the main concepts of metaheuristics, Section 4 outlines the proposed shadowed parameter adaptation approach, Section 5 summarizes the optimized controller and the achieved results, and Section 6 puts forward the conclusions and possible lines of future research.

2. Type-2 Fuzzy Systems and Shadowed Sets

Fuzzy logic has evolved over the years; it started with what is known as type-1 fuzzy logic in 1965 [45], and its main objective was to model the vagueness of the real world, as well as to use mathematical models to represent human knowledge. Subsequently, it was followed by what is now called type-2 fuzzy logic, which allows the handling of uncertainty, and to model a type-2 fuzzy set, Equation (1) is used:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x)) | \forall x \in X, \forall u \in J_x^u \subseteq [0, 1]\} \quad (1)$$

In the literature, there are some alternative ways to model type-2 fuzzy logic, such as geometric approximation [46] and representations using vertical slices [47] and horizontal slices [48]. In this work, the main idea is focused on the use of generalized type-2 fuzzy modeling through the representation of α planes. Equation (2) represents the process by which the secondary axis of a generalized type-2 fuzzy system is discretized in several α planes. Each α plane can be calculated as an interval type-2 fuzzy system [49] and thus with the union of each plane α , the modeling of a generalized type-2 fuzzy system is obtained, expressed in Equation (3).

$$\tilde{A}_\alpha = \{((x, u), \alpha) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (2)$$

$$\tilde{A} = \bigcup \tilde{A}_\alpha \quad (3)$$

The shadowed type-2 fuzzy inference system [50] arises in order to reduce the computational cost generated by the use of α -planes. This work focuses on the use of shadowed type-2 fuzzy inference systems with only two optimal α -planes. To define the precision of the use of the optimal α -planes, the theory proposed by Pedrycz is used [51].

Equation (4) describes the intervals used to understand the definition of the shadowed sets, which consist of performing two α -cuts on a type-1 fuzzy set; the values are defined in the variables α and β , as shown below:

$$S_{\mu_A}(x) = \begin{cases} 1, & \text{if } \mu_A(x) \geq \alpha \\ 0, & \text{if } \mu_A(x) \leq \beta \\ [0, 1], & \text{if } \alpha \leq \mu_A(x) \leq \beta \end{cases} \quad (4)$$

Equation (5) was proposed in the literature by Pedrycz, in which certain optimal values are established for α and β , and the expression describes three regions, which are contained in $[0, 1]$ and their names are included in the equation.

$$\text{elevated area}_{(\alpha, \beta)}(\mu_A) + \text{reduced area}_{(\alpha, \beta)}(\mu_A) = \text{shadowed area}_{(\alpha, \beta)}(\mu_A) \quad (5)$$

An image depiction of Equation (5) is illustrated in Figure 1.

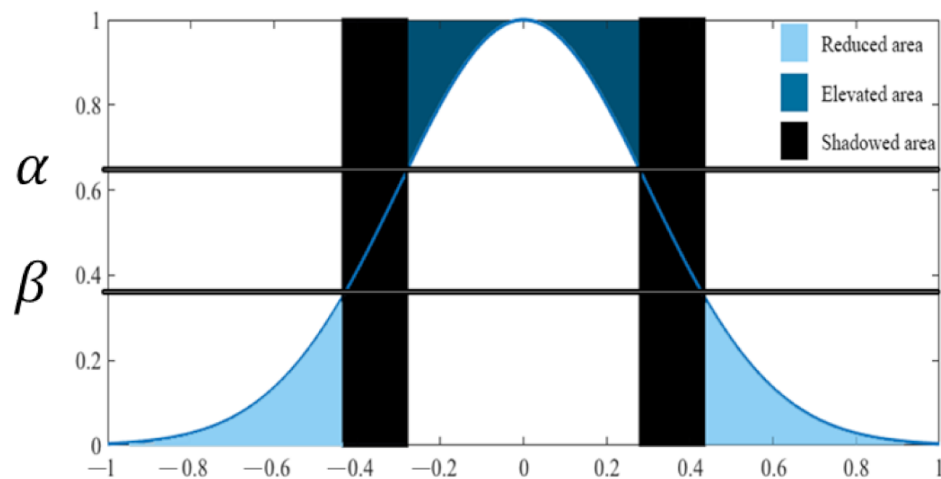


Figure 1. Shadowed set representation.

Based on Equation (6), the optimal values of α and β are obtained, and this requires an optimization process to be able to determine these values.

$$V(\alpha, \beta) = \left| \int_{x \in A_r} \mu_A(x) dx + \int_{x \in A_e} (1 - \mu_A(x)) dx - \int_{x \in S} dx \right| \quad (6)$$

This proposed shadowed type-2 FIS methodology is used for the dynamic adaptation of parameters of the metaheuristic algorithms in order to optimize the results obtained in the problem to be used. A modification to the shadowed type-2 fuzzy inference system (FIS) was made by Linda and Manic [52], who indicated that the secondary axis of the generalized type-2 fuzzy system uses the shadowed sets to find the optimal α and β values, and thus use those values to find the α planes.

Trapezoidal ST2 MF

There are different types of membership functions that can be utilized in fuzzy systems. In this work, a decision was made, based on simplicity, to use the Trapezoidal (TrapG) membership function, in which the main characteristic is that it resembles the trapezoidal GT2 membership function, which is composed of a Gaussian-type function as a secondary membership function. Equation (7) contains the parameter values that were used for the membership function, and Figure 2 illustrates the graphical representation of the membership function.

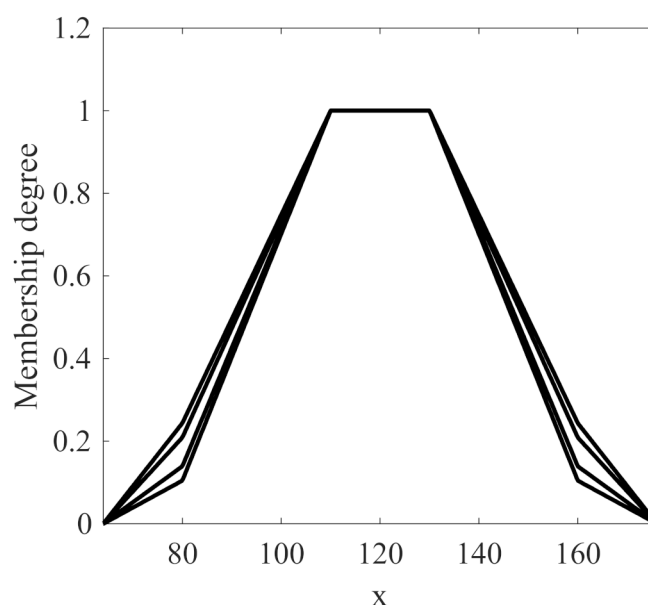


Figure 2. Trapezoidal ST2 membership function.

$$\text{TrapG ST2 MF} = \begin{cases} \alpha_o \left\{ \begin{array}{l} \bar{\mu}_O = \frac{\bar{\mu}_t(x) + \underline{\mu}_t(x)}{2} - 1.449 \\ \underline{\mu}_O = \frac{\bar{\mu}_t(x) + \underline{\mu}_t(x)}{2} + 1.449 \end{array} \right. \left| \begin{array}{l} \frac{\bar{\mu}_t(x) - \underline{\mu}_t(x)}{10} \\ \frac{\bar{\mu}_t(x) - \underline{\mu}_t(x)}{10} \end{array} \right. \\ \alpha_I \left\{ \begin{array}{l} \bar{\mu}_I = \frac{\bar{\mu}_t(x) + \underline{\mu}_t(x)}{2} - 0.9282 \\ \underline{\mu}_I = \frac{\bar{\mu}_t(x) + \underline{\mu}_t(x)}{2} + 0.9282 \end{array} \right. \left| \begin{array}{l} \frac{\bar{\mu}_t(x) - \underline{\mu}_t(x)}{10} \\ \frac{\bar{\mu}_t(x) - \underline{\mu}_t(x)}{10} \end{array} \right. \end{cases} \quad (7)$$

With the goal of analyzing the importance of implementing a shadowed fuzzy logic system to solve real problems, the following Table 1 highlights some interesting works that use some implementation or variants of type-2 fuzzy logic systems.

Table 1. Outline of some interesting works based on type-2 fuzzy logic systems improvements.

Recent Studies/Authors	Method Used	Objective of the Article	Study Case	Meta-Heuristics Used
Proposed Method	Shadowed Type-2 Fuzzy Systems	Find optimal parameters in the studied algorithms.	Fuzzy controllers: Cruise control	Harmony Search and Differential Evolution
Castillo O. et al. [2]	Shadowed type-2 fuzzy systems	Find an optimal parameters of the studied algorithms.	Fuzzy controller: DC motor	Harmony Search and Differential Evolution
Mohammadzadeh, A. et al. [5]	Type-3 Fuzzy Control	Find an optimal dynamic model in a period of time short and the control signal is designed based on this model.	Nonlinear systems: DC motors	Not Applied
Bernal, E. et al. [7]	Type-2 Fuzzy Logic System	Finding the best method that generates an optimal vector of values for the membership function.	Fuzzy controller: path tracking of an autonomous mobile robot	firefly algorithm and galactic swarm optimization
Türk, S. [17]	Interval type-2 Fuzzy Sets	Find the best configuration of the parameters of the interval type-2 membership functions for locating the electric charging stations.	Applied to a real-world public transport problem	Not Applied
Ochoa, P. et al. [43]	Interval Type-2 Fuzzy Logic System	Find an optimal parameters of the algorithms studied.	Fuzzy controllers	Differential Evolution Algorithm

Table 1. Cont.

Recent Studies/Authors	Method Used	Objective of the Article	Study Case	Meta-Heuristics Used
Wagner, C. et al. [47]	General Type-2 Fuzzy Logic Systems Based on zSlices	Reduction in the computational time with high levels of uncertainties.	Fuzzy controller: a two-wheeled mobile robot	Not Applied
Mendel, J.M. et al. [49]	Alpha-Plane Representation for Type-2 Fuzzy Sets	Find the precision to forecast noise-corrupted measurements.	Time series: a chaotic Mackey-Glass	Not Applied
Pedrycz, W. [50]	Shadowed Sets	Demonstrate how shadowed sets help in problems in data interpretation in fuzzy clustering.	Fuzzy clustering	Not Applied
Wijayasekara, D. et al. [52].	Shadowed Type-2 Fuzzy Logic Systems	Outlines a novel concept of shadowed type-2 fuzzy logic systems (ST2 FLS).	Explanation of important theory	Not Applied

3. Metaheuristic Algorithms

At the present time, the use of metaheuristic algorithms for solving problems in different areas is very common, these algorithms are used in order to optimize the results of the problems in which they are used. For this work, two algorithms are presented as well as their main operating characteristics and the equations involved in each of the algorithms for its implementation. The algorithms are: harmony search [53] and differential evolution [54]. Figure 3 illustrates the implementation in the control problem with the proposed use of shadowed type-2 fuzzy logic systems.

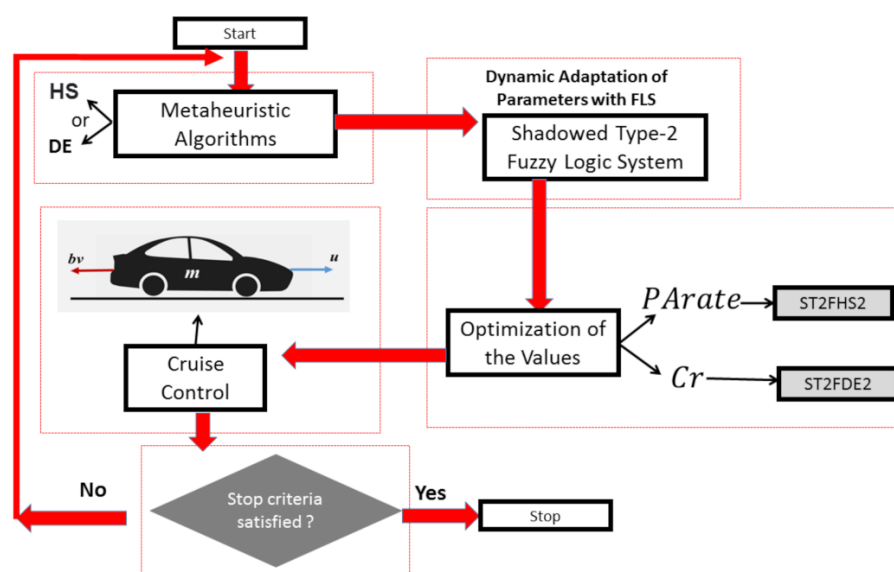


Figure 3. Proposed method.

3.1. Harmony Search Algorithm

In the recent state of the art, there is a large number of metaheuristics, but this research focuses on the study of an algorithm that was created by Zong Woo Geem in 2001 [55]. In particular, this algorithm was inspired by musical composition imitating the behavior of an orchestra, and it has been implemented in its mathematical form using Equations (8)–(10) for the resolution of different problems in the real world.

$$HMR \in [0, 1] \quad (8)$$

$$X_{new} = X_{old} + bp(2rand - 1) \quad (9)$$

$$P_{Arate} = P_{Lower\ limit} + P_{Range} * rand$$

$$\text{Where } P_{Range} = P_{Upper\ Limit} - P_{Lower\ Limit} \quad (10)$$

Using the *HMR* parameter, the HS algorithm performs memory consideration if $0 \leq rand \leq HMR$ and random selection if $HMR \leq rand \leq 1$. Based on the *PARate* parameter, HS performs pitch adjustment if $0 \leq rand \leq PARate$ and no change is made if $PARate \leq rand \leq 1$. The memory consideration has the following meaning: X_{new} comes from HM. The random selection has the following meaning: X_{new} comes from the total value range. The pitch adjustment has the following meaning: X_{new} is further tweaked by adding delta (a small positive or negative value; if we want to use *bp*, we have to establish it first) once X_{new} is obtained from the memory consideration.

3.2. Differential Evolution Algorithm

Differential evolution [56] has been utilized in different optimization areas, and this work is no exception; the algorithm is relatively simple and only consists of four stages in its development.

Population initialization is the first step to start the DE algorithm, which is composed of *N*-dimensional vectors and is defined by Equations (11)–(16):

$$P_{x,g} = (x_{i,g}), i = 0, 1, \dots, Np - 1, g = 0, 1, \dots, g_{max} \quad (11)$$

$$x_{i,g} = (x_{j,i,g}), j = 0, 1, \dots, D - 1 \quad (12)$$

$$P_{v,g} = (v_{i,g}), i = 0, 1, \dots, Np - 1, g = 0, 1, \dots, g_{max} \quad (13)$$

$$v_{i,g} = (v_{j,i,g}), j = 0, 1, \dots, D - 1 \quad (14)$$

$$P_{u,g} = (u_{i,g}), i = 0, 1, \dots, Np - 1, g = 0, 1, \dots, g_{max} \quad (15)$$

$$u_{i,g} = (u_{j,i,g}), j = 0, 1, \dots, D - 1 \quad (16)$$

After creating the vectors with which the algorithm starts to work, we have to define the search space in which these vectors have an upper limit and a lower limit; this will depend on the problem to be used. This process is called initialization, and this is defined by Equation (17):

$$x_{j,i,0} = rand_j(0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L} \quad (17)$$

More specifically, the differential mutation adds a random component to DE and the following equation illustrates how to mix three different vectors chosen randomly to create a mutant vector; this is shown in Equation (18):

$$v_{i,g} = x_{r_0,g} + F \cdot (x_{r_1,g} - x_{r_2,g}) \quad (18)$$

To complement the effect of the differential mutation search strategy, DE also uses uniform crossover. More specifically, DE crosses each vector with a mutant vector defined by Equation (19):

$$u_{i,g} = u_{j,i,g} \begin{cases} v_{j,i,g} & \text{if } (rand_j(0,1) \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (19)$$

The next step in the algorithm is to make the selection; in this process, a comparison is made between the mutated vector and the objective function. Equation (20) shows the selection process:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases} \quad (20)$$

The stopping criterion for the algorithm can occur when an optimal solution is found or when the termination criteria specified at the beginning of the algorithm are satisfied.

4. Dynamic Parameter Adaptation

For the present work, the two abovementioned algorithms (HS and DE) were considered for the dynamic parameter adaptation. A fuzzy system was designed and imple-

mented for each of the algorithms, since each algorithm has a specific parameter to be dynamically moved.

The structure of the fuzzy systems was formed by one input and one output, and the specifications for each of the algorithms are separately shown below.

For the use of the HS algorithm, the iterations represent the input to the fuzzy system and the output is $PArate$, while for the DE algorithm, the generations represent the input, while Cr (crossover) represents the output.

The input of each of the fuzzy systems is calculated using Equation (21), where *Current Test* represents the current iteration or generation in the algorithm, which is divided by the maximum number of iterations or generations, which is called *Maximum number of tests*.

$$Test = \frac{Current\ Test}{Maximum\ number\ of\ tests} \quad (21)$$

Equations (22) and (23) express the outputs of each of the HS and DE algorithms, respectively.

$$PArate = \frac{\sum_{i=1}^{r_{PArate}} \mu_i^{PArate} (PArate_{1i})}{\sum_{i=1}^{r_{PArate}} \mu_i^{PArate}} \quad (22)$$

$PArate$ symbolizes the pitch adjustment; r_{PArate} symbolizes the number of rules of the shadowed type-2 system that correspond to $PArate$; $PArate_{1i}$ symbolizes the output result for rule i with respect to $PArate$; μ_i^{PArate} is the membership function of rule i that corresponds to $PArate$.

$$Cr = \frac{\sum_{i=1}^{r_{Cr}} \mu_i^{Cr} (Cr_{1i})}{\sum_{i=1}^{r_{Cr}} \mu_i^{Cr}} \quad (23)$$

where Cr is the crossover; r_{Cr} is the number of rules of the shadowed type-2 system that correspond to Cr ; Cr_{1i} is the output result for rule i that corresponds to Cr ; μ_i^{Cr} is the membership function of rule i that corresponds to Cr .

The combination of the shadowed type-2 system with the HS algorithm is called ST2FHS2 in the simulations carried out in this work, and the combination of the shadowed type-2 system with the DE algorithm is called ST2FDE2.

Each fuzzy system has a structure of one input and one output. In this case, the input was composed of three membership functions as well as the output. The graphical representation of the input and output can be seen in Figures 4 and 5, respectively. The names of each of the membership functions are: Basic, Intermediate and Advanced, and are represented in Figure 6.

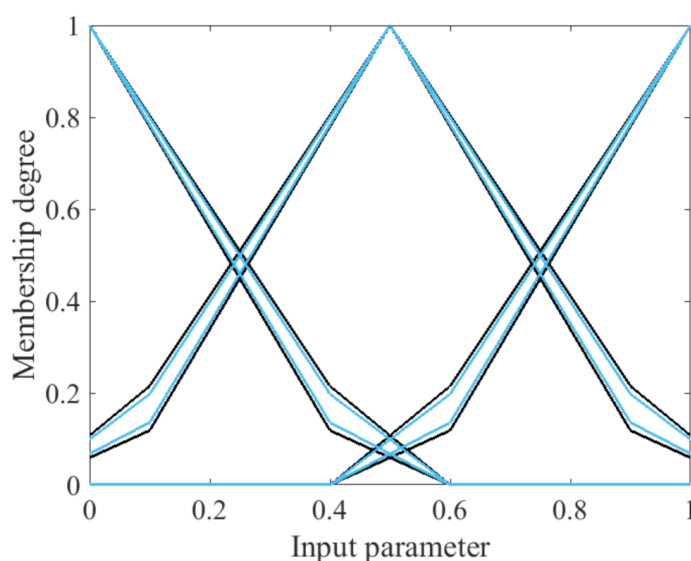


Figure 4. Linguistic input parameter membership functions.

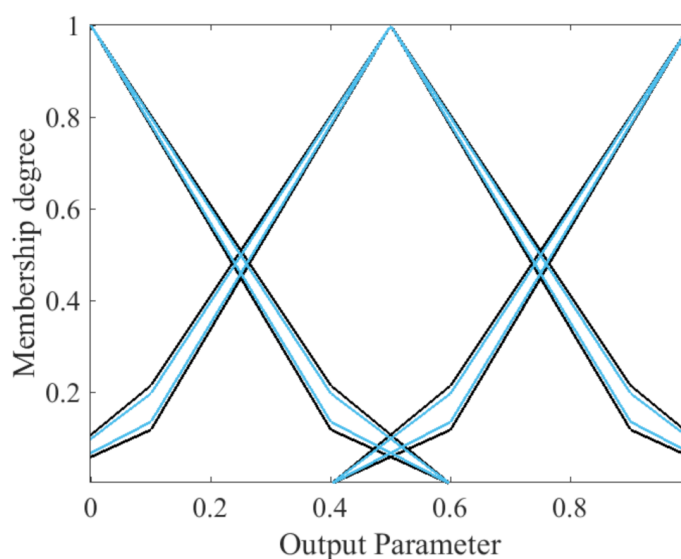


Figure 5. Linguistic output parameter membership functions.

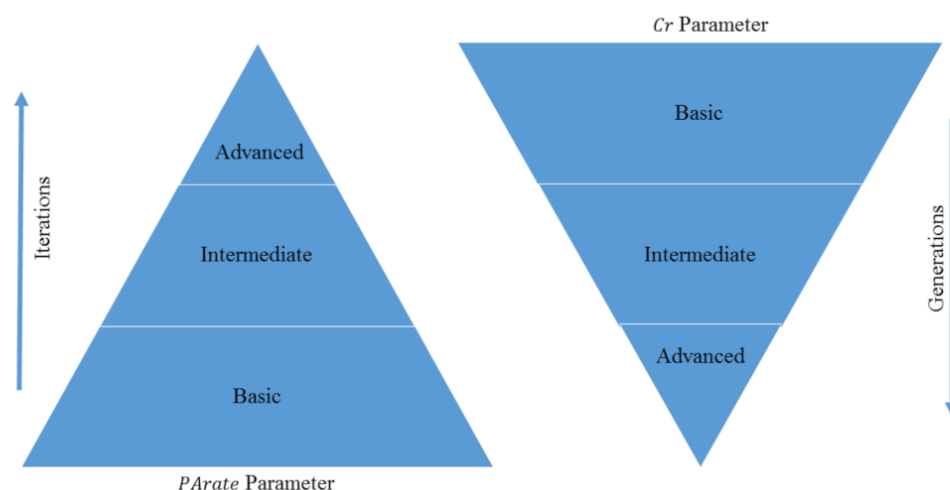


Figure 6. Fuzzy rules for both methods.

5. Controllers Optimization

Feedback control problems are very popular for testing the performance of an algorithm; therefore, this article uses a control case that is widely used in proportional integral (PI) and proportional integral derivative (PID) controllers, and the implementation in this work was carried out in a type-1 fuzzy controller, with the goal of validating the performance of the two proposed methods.

5.1. Methodology for Controller Optimization

This section explains in detail the implementation of a type-1 fuzzy controller to the cruise control system model. The cruise control controller is made up of trapezoidal and triangular membership functions, both at its two inputs and at its output. Equations (24) and (25) show the mathematical basis for the construction of each of these membership functions. In Table 2, the parameter values assigned to each linguistic variable inputs and the output are presented.

$$\text{trapezoidalmf}(x; a, b, c, d) = \max(\min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0) \quad (24)$$

$$\text{triangularmf}(x; a, b, c) = \max(\min(\frac{x-a}{b-a}, \frac{c-x}{c-b}), 0) \quad (25)$$

Table 2. Type-1 fuzzy system parameter values.

Input Err				
MF	a	b	c	d
−V	−1	−1	−0.5	0
ZV	−0.5	0	0.5	− − −
+V	0	0.5	1	1
Input Change Err				
E−	−1	−1	−0.4	−0.1
E−M	−0.4	−0.2	0	− − −
ZE	−0.01	0	0.10	− − −
E+M	0	0.2	0.4	− − −
E+	0.1	0.4	1	1
Output Voltage				
D	−1	−1	−0.6	−0.09
D+	−0.4	−0.2	0	− − −
M	−0.1	0	0.1	− − −
A+	0	0.2	0.4	− − −
A	0.09	0.6	1	1

The structure of the type-1 fuzzy system for the controller is presented in Figure 7, the rules are presented in Table 3, and the control surface can be seen in Figure 8.

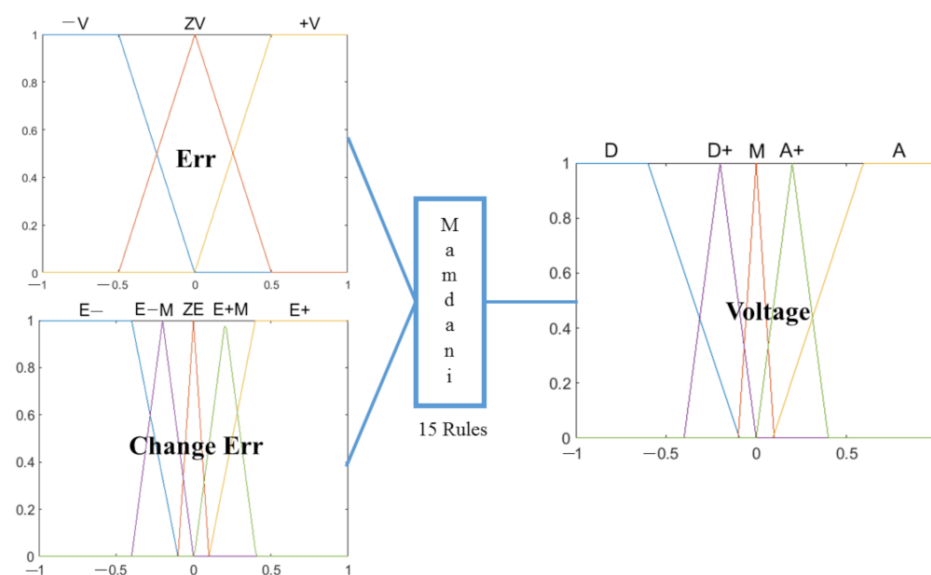
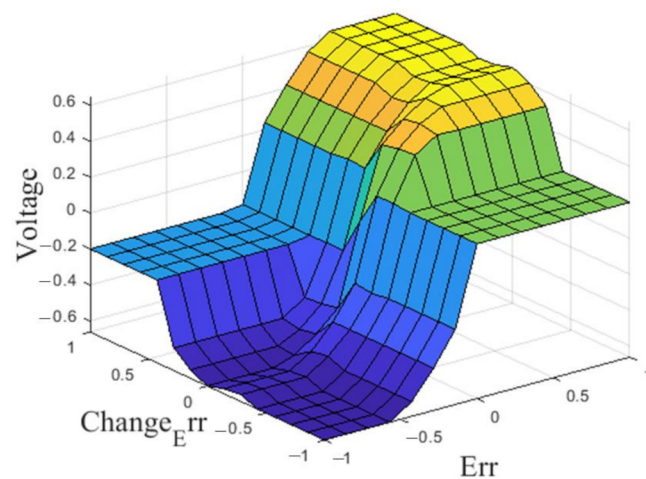
**Figure 7.** General structure of the type-1 fuzzy system.**Figure 8.** Cruise control surface.

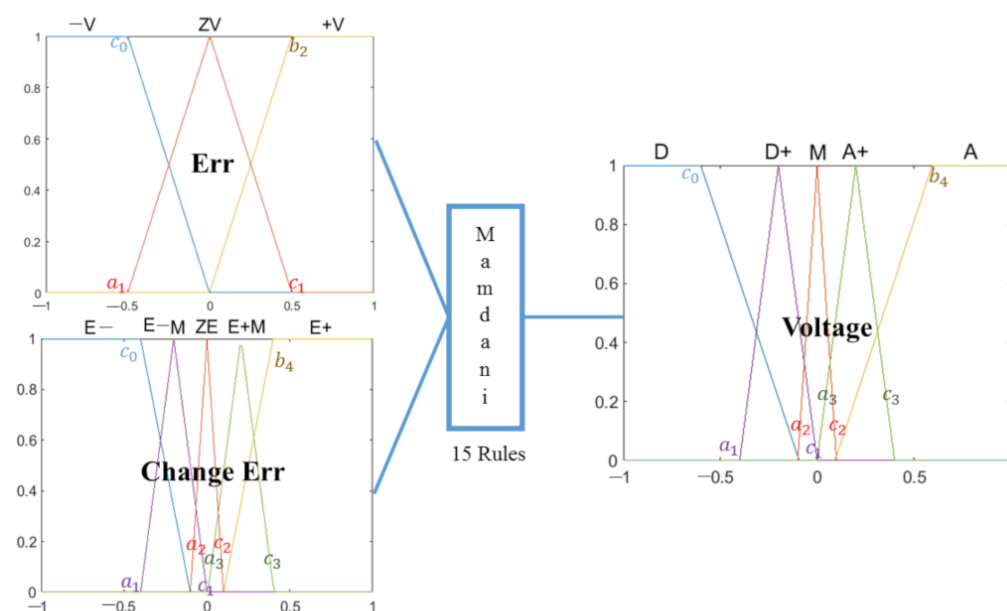
Table 3. Fuzzy rules for the cruise control controller.

Rules	Inputs		Output
	Err	Change Err	Voltage
1	−V	E	D
2	−V	ZE	D
3	−V	E+	D+
4	Zv	E	A+
5	ZV	E+	D+
6	+V	E	A+
7	+V	ZE	A
8	+V	E+	A
9	ZV	ZE	M
10	−V	EM	D
11	ZV	EM	A+
12	+V	EM	A
13	+V	E+M	A
14	ZV	E+M	D+
15	−V	E+M	D

The two methods proposed in this article were implemented to carry out the optimization of the values of the membership functions of the type-1 fuzzy system implemented in the cruise control controller. In order to calculate the RMSE error of control, Equation (26) is used.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \quad (26)$$

Figure 9 depicts the individuals in each of the membership functions of the cruise control controller.

**Figure 9.** Representation of the individuals for the cruise controller.

The type-1 system implemented for this controller contains a total of 45 points, where 25 points are fixed and 20 are optimized with each proposed method. The limits of the values for each linguistic variable for each input and output are illustrated in Table 4.

Table 4. Limitations of the linguistic values for the cruise controller.

Err	Change Err	Voltage
$-V$ $a_0 = b_0 = -1$ $-1 < c_0 < -0.5$ $d_0 = 0$	$E-$ $a_0 = b_0 = -1$ $-1 < c_0 < -0.4$ $d_0 = -0.1$	D $a_0 = b_0 = -1$ $-1 < c_0 < -0.6$ $d_0 = -0.09$
ZV $-1 < a_1 < 0$ $b_1 = 0$ $0 < c_1 < 1$	$E-M$ $-1 < a_1 < -0.1$ $b_1 = -0.2$ $-0.1 < c_1 < 0$	$D+$ $-1 < a_1 < -0.4$ $b_1 = -0.2$ $-0.2 < c_1 < 0$
$+V$ $a_2 = 0$ $0 < b_2 < 0.5$ $c_2 = d_2 = 1$	ZE $-0.09 < a_2 < 0$ $b_2 = 0$ $0 < c_2 < 0.10$	M $-0.1 < a_2 < 0$ $b_2 = 0$ $0 < c_2 < 0.1$
	$E+M$ $0 < a_3 < 0.10$ $b_3 = 0.2$ $0.1 < c_3 < 1$	$A+$ $0 < a_3 < 0.1$ $b_3 = 0.2$ $0.09 < c_3 < 1$
	$E+$ $a_4 = 0.1$ $0.2 < b_4 < 0.4$ $c_4 = d_4 = 1$	A $a_4 = 0.09$ $0.2 < b_4 < 0.6$ $c_4 = d_4 = 1$

5.2. Cruise Control

Many modern vehicles have a feedback control system, the automatic cruise control system, which is a relatively simple device. The aim of the cruise control system to maintain a constant speed of the vehicle without being affected by external perturbations, such as the slope of the road or changes in the wind. To fulfill the main objective of the controller, the vehicle speed must be measured, comparing it with the reference speed and then adjusting the throttle based on a control strategy. Figure 10 represents the model of the vehicle dynamics, where the vehicle mass is actuated by a control force. For this controller, the parameters of the system are represented in Table 5.

**Figure 10.** Cruise controller representation.**Table 5.** The system parameters of the cruise controller.

Parameter	Description	Value
m	Vehicle mass	1000 kg
b	Damping coefficient	50 N·s/m
r	Reference speed	10 m/s
u	Force on the vehicle	500 N

5.3. Experiments

To check the performance of the proposed methods, a type-1 fuzzy system was implemented to a feedback control problem. In this case, the control problem is called cruise control. The main goal of this controller is to maintain a constant speed of 10 m/s in the vehicle regardless of the disturbances that may arise. All the experimentation in this work was developed in MATLAB [57].

Experiments were carried out without applying noise to the controller and applying a Gaussian noise of 0.5 percent, which represents 50 percent of the noise in the controller.

The results obtained for the ST2FHS2 and ST2FDE2 methods are shown in Tables 6 and 7, respectively. In these tables, the values that are analyzed (the minimum, maximum, average and standard deviation obtained from each method in 30 experiments) are presented. Just as the Z-value shown in Equation (27), which was obtained by performing the test statistic Z-test, the values of the parameters used in this test were as follows: level of significance 0.05, for all Z-values lower than -1.645 .

$$Z = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}} \quad (27)$$

Table 6. Results for the ST2FHS2 method.

Method	HS-FLC without Noise	ST2FHS2-FLC without Noise	HS-FLC with Noise	ST2FHS2-FLC with Noise
Best	5.98×10^{-1}	6.60×10^{-2}	3.58×10^{-1}	4.43×10^{-2}
Worst	$9.17 \times 10^{+0}$	$9.19 \times 10^{+0}$	$6.64 \times 10^{+0}$	$6.29 \times 10^{+0}$
Average	$4.64 \times 10^{+0}$	$3.36 \times 10^{+0}$	$4.40 \times 10^{+0}$	$3.58 \times 10^{+0}$
Std.	$3.25 \times 10^{+0}$	$2.29 \times 10^{+0}$	$1.84 \times 10^{+0}$	$1.85 \times 10^{+0}$
Z-value	-1.7634		-1.7213	

Table 7. Results for the ST2FDE2 method.

Method	DE-FLC without Noise	ST2FDE2-FLC without Noise	DE-FLC with Noise	ST2FDE2-FLC with Noise
Best	9.28×10^{-1}	7.59×10^{-2}	3.00×10^{-1}	7.05×10^{-2}
Worst	$5.17 \times 10^{+0}$	$1.76 \times 10^{+0}$	$5.13 \times 10^{+0}$	$1.15 \times 10^{+0}$
Average	$6.04 \times 10^{+0}$	$5.46 \times 10^{+0}$	$5.11 \times 10^{+0}$	$4.15 \times 10^{+0}$
Std.	$1.36 \times 10^{+0}$	6.79×10^{-1}	6.79×10^{-1}	1.45×10^{-1}
Z-value	-2.0899		-7.5732	

Figure 11 shows the best simulation obtained with the HS and ST2FHS2 methods without noise in the controller and Figure 12 presents the best surface obtained from each of the aforementioned methods.

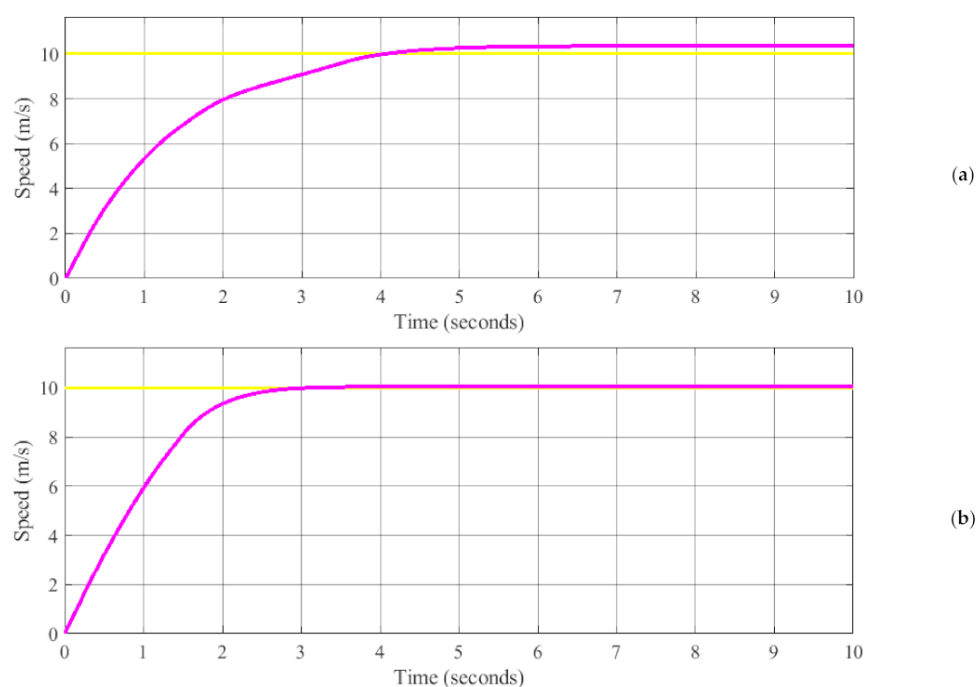


Figure 11. Comparison of the results obtained optimizing controller cruise control without noise: (a) using the HS method and (b) using the proposed ST2FHS2 method.

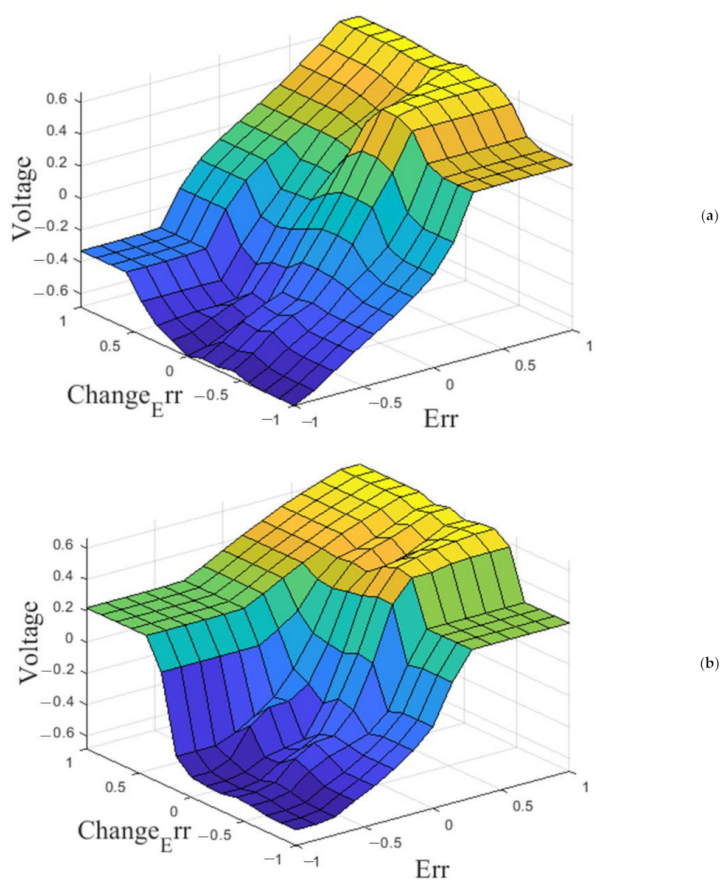


Figure 12. Comparison of the surface obtained optimizing controller cruise control without noise: (a) using the HS method and (b) using the proposed ST2FHS2 method.

Figure 13 shows the best simulation obtained with the HS and ST2FHS2 methods with noise in the controller and Figure 14 presents the best surface obtained from each of the aforementioned methods.

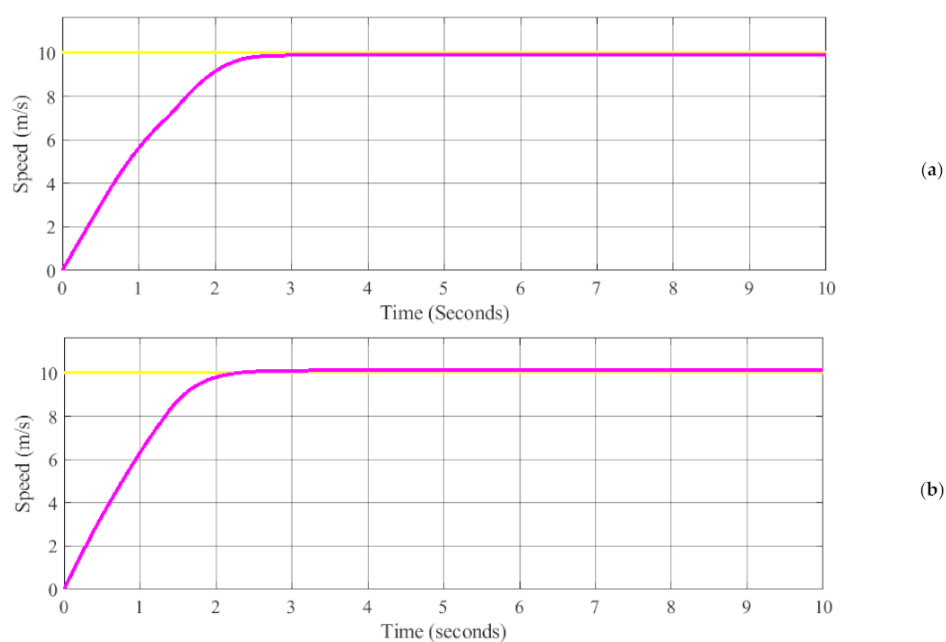


Figure 13. Comparison of the results obtained optimizing controller cruise control with noise: (a) using the HS method and (b) using the proposed ST2FHS2 method.

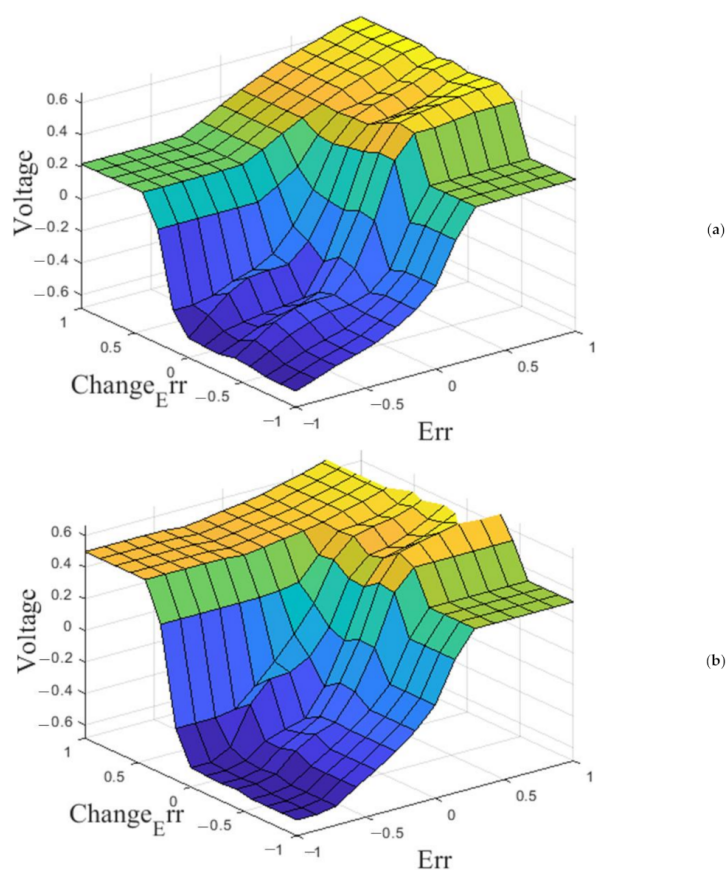


Figure 14. Comparison of the surface obtained optimizing controller cruise control with noise: (a) using the HS method and (b) using the proposed ST2FHS2 method.

The yellow lines in Figures 11 and 13 indicate the controller reference and the pink line indicates the value obtained from the best simulation found with each method.

Figure 15 shows the best simulation obtained with the DE and ST2FDE2 methods without noise in the controller and Figure 16 presents the best surface obtained from each of the aforementioned methods.

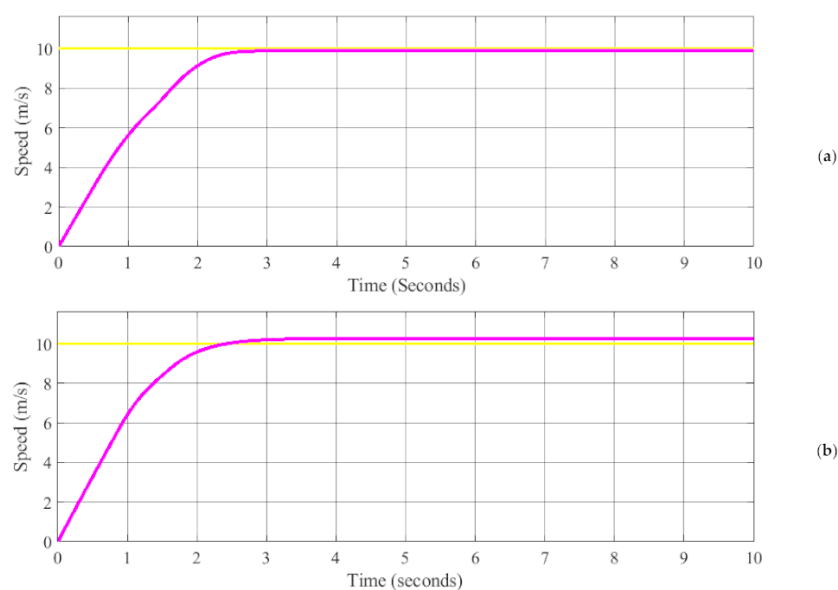


Figure 15. Comparison of the results obtained optimizing controller cruise control without noise: (a) using the DE method and (b) using the proposed ST2FDE2 method.

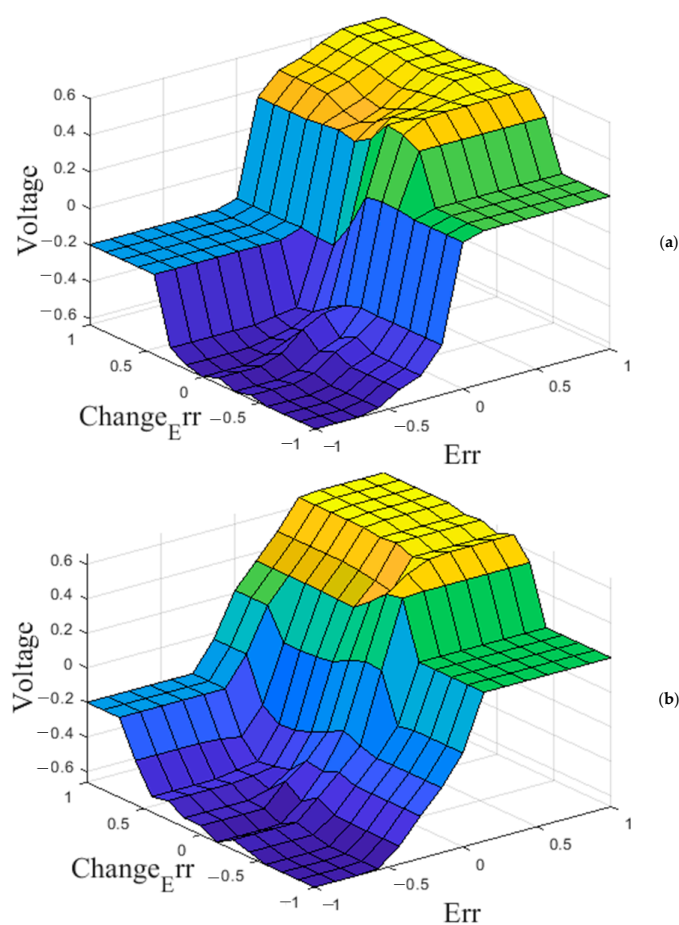


Figure 16. Comparison of the surface obtained optimizing controller cruise control without noise: (a) using the DE method and (b) using the proposed ST2FDE2 method.

Figure 17 shows the best simulation obtained with the DE and ST2FDE2 methods with noise in the controller and Figure 18 presents the best surface obtained from each of the aforementioned methods.

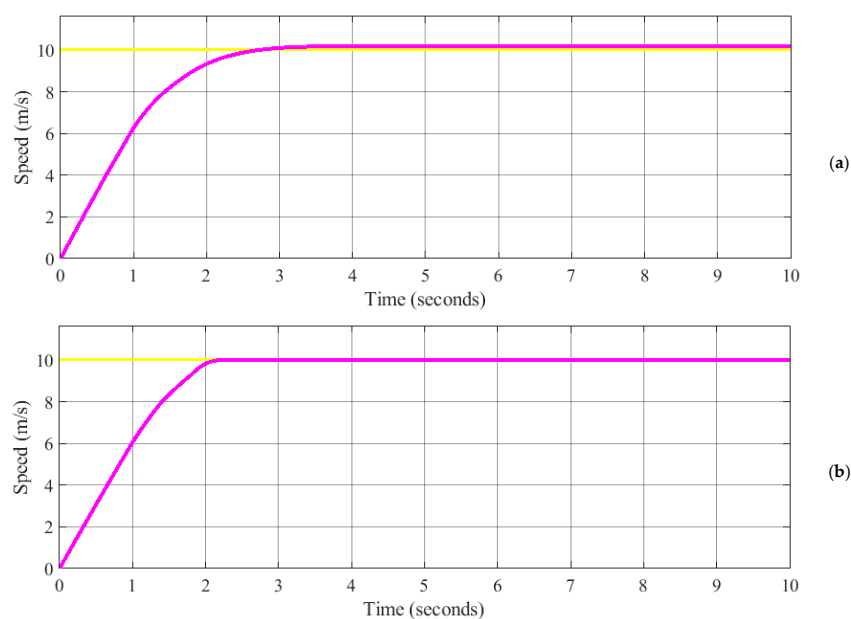


Figure 17. Comparison of the results obtained optimizing controller cruise control with noise: (a) using the DE method and (b) using the proposed ST2FDE2 method.

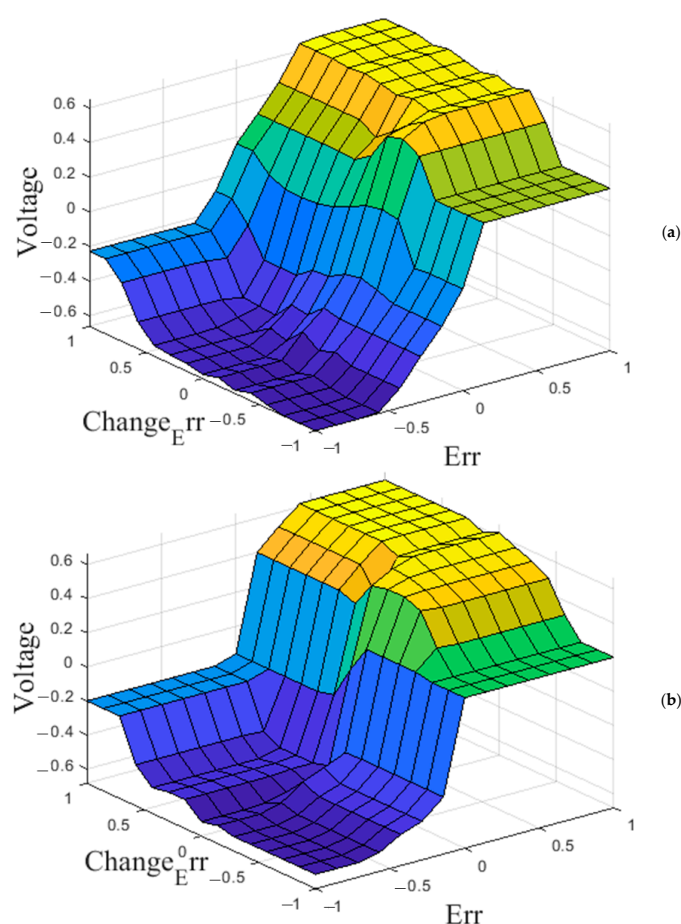


Figure 18. Comparison of the surface obtained optimizing controller cruise control with noise: (a) using the DE method and (b) using the proposed ST2FDE2 method.

The yellow line in Figures 15 and 17 indicates the controller reference and the pink line indicates the value obtained from the best simulation found with each method.

6. Conclusions

The most important contribution of this article is the implementation of shadowed type-2 fuzzy logic in the HS and DE algorithms for dynamically moving the parameters of $PARate$ and Cr , respectively, as these parameters had not been previously dynamically moved for the solution of the above control problem. For this work, the control problem in which the testing was performed was also solved separately by each algorithm used.

For the HS algorithm, the analysis of the results shows that when comparing HS-FLC and ST2FHS2-FLC both without noise, the best results are 5.98×10^{-1} and 6.60×10^{-2} , respectively. It is clearly observed that there is a difference, and the statistical test shows that ST2FHS2-FLC without noise is lower on average than HS-FLC.

In the comparison of HS-FLC and ST2FHS2-FLC FLC with noise, the best results obtained are 3.58×10^{-1} and 4.43×10^{-2} , respectively, and in this paper, it is validated that the best result is ST2FHS2-FLC with noise. This comparison was verified statistically, proving that ST2FHS2-FLC with noise is better than HS-FLC.

Comparing DE-FLC and ST2FDE2-FLC without noise, the best results were 9.28×10^{-1} and 7.59×10^{-2} , respectively, and we can note that there is a difference between both results. The statistical test shows that ST2FDE2-FLC is better on average than DE-FLC without noise. The comparison of the algorithm using a noise level in the controller shows that the best results obtained for DE-FLC with noise is 3.00×10^{-1} and for ST2FDE2-FLC with noise is 7.05×10^{-2} ; so, it is statistically possible to confirm that ST2FDE2-FLC with noise is better on average compared to DE-FLC with noise.

In general, the finding presented in this paper is that the proposal to adapt the $PARate$ and Cr parameters for HS and DE algorithms combined with shadowed type-2 fuzzy logic produces a significant improvement over the original algorithms and demonstrates that the use of the proposed approach shows good results even when adding different noise levels to the controller. As future work, we envision the application of the proposed approach for the optimal design of fuzzy controllers for more complex problems, and also for different areas of application, such as image processing, control to the path of robots in simulations, or medical diagnosis.

Regarding the study of the shadowed type-2 fuzzy logic systems, one relevant conclusion is that the main limitation is the computer performance of the metaheuristic, because it is very important for the improvement of the efficiency of this method in its application. The main contribution in this work was that it put forward a way of reducing execution time, with the main goal to obtain good results. In this case, a control problem based on precision is implemented, which is the cruise control problem. This case study requires good stabilization to achieve the control goal in this problem.

In regard to future works, it would be interesting to observe and analyze the execution time and performance of the shadowed type-2 fuzzy logic systems when higher noise levels are considered in the control process.

Author Contributions: Conceptualization, creation of main idea, writing—review and editing, C.P., P.O. and O.C.; formal analysis, O.C. and Z.W.G.; methodology, P.M. and validation, L.A.-A. and Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Energy Cloud R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (2019M3F2A1073164). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (2020R1A2C1A01011131).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the program of the Division of Graduate Studies and Research of Tijuana Institute of Technology, specifically Oscar Castillo and Patricia Melin, program coordinator, before being interested in our research and for creating excellent teamwork in collaboration with all the co-authors on this paper.

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

References

1. Castillo, O.; Valdez, F.; Soria, J.; Yoon, J.H.; Geem, Z.W.; Peraza, C.; Amador-Angulo, L. Optimal Design of Fuzzy Systems Using Differential Evolution and Harmony Search Algorithms with Dynamic Parameter Adaptation. *Appl. Sci.* **2020**, *10*, 6146. [\[CrossRef\]](#)
2. Castillo, O.; Melin, P.; Valdez, F.; Soria, J.; Ontiveros-Robles, E.; Peraza, C.; Ochoa, P. Shadowed type-2 fuzzy systems for dynamic parameter adaptation in harmony search and differential evolution algorithms. *Algorithms* **2019**, *12*, 17. [\[CrossRef\]](#)
3. Castillo, O.; Valdez, F.; Soria, J.; Amador-Angulo, L.; Ochoa, P.; Peraza, C. Comparative study in fuzzy controller optimization using bee colony, differential evolution, and harmony search algorithms. *Algorithms* **2019**, *12*, 9. [\[CrossRef\]](#)
4. Castillo, O.; Melin, P.; Ontiveros, E.; Peraza, C.; Ochoa, P.; Valdez, F.; Soria, J. A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. *Eng. Appl. Artif. Intell.* **2019**, *85*, 666–680. [\[CrossRef\]](#)
5. Mohammadzadeh, A.; Castillo, O.; Band, S.S.; Mosavi, A. A Novel Fractional-Order Multiple-Model Type-3 Fuzzy Control for Nonlinear Systems with Unmodeled Dynamics. *Int. J. Fuzzy Syst.* **2021**, 1–19. [\[CrossRef\]](#)
6. Castillo, O.; Valdez, F.; Peraza, C.; Yoon, J.H.; Geem, Z.W. High-Speed Interval Type-2 Fuzzy Systems for Dynamic Parameter Adaptation in Harmony Search for Optimal Design of Fuzzy Controllers. *Mathematics* **2021**, *9*, 758. [\[CrossRef\]](#)
7. Bernal, E.; Lagunes, M.L.; Castillo, O.; Soria, J.; Valdez, F. Optimization of type-2 fuzzy logic controller design using the GSO and FA algorithms. *Int. J. Fuzzy Syst.* **2021**, *23*, 42–57. [\[CrossRef\]](#)
8. Mohammedali, M.N.; Rasheed, M.; Shihab, S.; Rashid, T.; Hamed, S.H.A. Optimal Parameters Estimation of Silicon Solar Cell Using Fuzzy Logic: Analytical Method. *J. Al-Qadisiyah Comput. Sci. Math.* **2021**, *13*, 22.

9. Yadav, R.S. Application of soft computing techniques to calculation of medicine dose during the treatment of patient: A fuzzy logic approach. In *Handbook of Computational Intelligence in Biomedical Engineering and Healthcare*; Academic Press: Cambridge, MA, USA, 2021; pp. 151–178.
10. Hossain, M.A.; Hossain, M.S. Mathematical Approach to Identify Coronavirus Disease (COVID-19) Using Fuzzy Logic Inference System. *Int. J. Fuzzy Math. Arch.* **2021**, *19*, 37–47.
11. Sandal, B.; Hacıoglu, Y.; Salihoglu, Z.; Yagiz, N. Fuzzy Logic Preanesthetic Risk Evaluation of Laparoscopic Cholecystectomy Operations. *Am. Surg.* **2021**, 1–10. [\[CrossRef\]](#)
12. Choudhury, B.S.; Dhara, P.S.; Saha, P. An application of fuzzy logic on importing medicines. *Int. J. Healthc. Manag.* **2021**, *14*, 456–461. [\[CrossRef\]](#)
13. Khalil, J.; Labib, A.W. On the optimization of maintenance storage cost in industry a fuzzy logic application. *Int. J. Qual. Reliab. Manag.* **2021**. [\[CrossRef\]](#)
14. Kumar, P.M.; Babu, G.C.; Selvaraj, A.; Raza, M.; Luhach, A.K.; Díaz, V.G. Multi-criteria-based approach for job scheduling in industry 4.0 in smart cities using fuzzy logic. *Soft Comput.* **2021**, *25*, 1–16. [\[CrossRef\]](#)
15. Caiado, R.G.G.; Scavarda, L.F.; Gavião, L.O.; Ivson, P.; De Mattos Nascimento, D.L.; Garza-Reyes, J.A. A fuzzy rule-based industry 4.0 maturity model for operations and supply chain management. *Int. J. Prod. Econ.* **2021**, *231*, 107883. [\[CrossRef\]](#)
16. BİLGİÇ, H.; Kuvvetli, Y.; Baykal, P.D. Determination of Difficulty Level for Garment Model with Fuzzy Logic Method. *Tekst. Ve Mühendis* **2021**, *28*, 39–47. [\[CrossRef\]](#)
17. Türk, S.; Deveci, M.; Özcan, E.; Canitez, F.; John, R. Interval type-2 fuzzy sets improved by Simulated Annealing for locating the electric charging stations. *Inf. Sci.* **2021**, *547*, 641–666. [\[CrossRef\]](#)
18. Abbasi, M.; Abbasi, E.; Mohammadi-Ivatloo, B. Single and multi-objective optimal power flow using a new differential-based harmony search algorithm. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 851–871. [\[CrossRef\]](#)
19. Srikanth, R.; Bikshalu, K. Multilevel thresholding image segmentation based on energy curve with harmony Search Algorithm. *Ain Shams Eng. J.* **2021**, *12*, 1–20. [\[CrossRef\]](#)
20. Mahafzah, B.A.; Alshraideh, M. Hybrid harmony search algorithm for social network contact tracing of COVID-19. *Soft Comput.* **2021**, 1–23. [\[CrossRef\]](#)
21. Abualigah, L.; Diabat, A.; Geem, Z.W. A comprehensive survey of the harmony search algorithm in clustering applications. *Appl. Sci.* **2020**, *10*, 3827. [\[CrossRef\]](#)
22. Ahmed, S.; Ghosh, K.K.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Hybrid of harmony search algorithm and ring theory-based evolutionary algorithm for feature selection. *IEEE Access* **2020**, *8*, 102629–102645. [\[CrossRef\]](#)
23. Hasanipanah, M.; Keshtegar, B.; Thai, D.K.; Troung, N.T. An ANN-adaptive dynamical harmony search algorithm to approximate the flyrock resulting from blasting. *Eng. Comput.* **2020**, 1–13. [\[CrossRef\]](#)
24. Xin, Y.; Yi, J.; Zhang, K.; Chen, C.; Xiong, J. Offline selective harmonic elimination with (2N+1) output voltage levels in modular multilevel converter using a differential harmony search algorithm. *IEEE Access* **2020**, *8*, 121596–121610. [\[CrossRef\]](#)
25. Shiva, C.K.; Kumar, R. Quasi-oppositional harmony search algorithm approach for Ad Hoc and Sensor networks. In *Nature Inspired Computing for Wireless Sensor Networks*; Springer: Singapore, 2020; pp. 175–194.
26. M'zoughi, F.; Garrido, I.; Garrido, A.J.; De La Sen, M. Self-adaptive global-best harmony search algorithm-based airflow control of a wells-turbine-based oscillating-water column. *Appl. Sci.* **2020**, *10*, 4628. [\[CrossRef\]](#)
27. Tuo, S.; Liu, H.; Chen, H. Multipopulation harmony search algorithm for the detection of high-order SNP interactions. *Bioinformatics* **2020**, *36*, 4389–4398. [\[CrossRef\]](#)
28. Bae, J.H.; Kim, M.; Lim, J.S.; Geem, Z.W. Feature Selection for Colon Cancer Detection Using K-Means Clustering and Modified Harmony Search Algorithm. *Mathematics* **2021**, *9*, 570. [\[CrossRef\]](#)
29. Gholami, J.; Pourpanah, F.; Wang, X. Feature selection based on improved binary global harmony search for data classification. *Appl. Soft Comput.* **2020**, *93*, 106402. [\[CrossRef\]](#)
30. Agajie, T.F.; Khan, B.; Alhelou, H.H.; Mahela, O.P. Optimal expansion planning of distribution system using grid-based multi-objective harmony search algorithm. *Comput. Electr. Eng.* **2020**, *87*, 106823. [\[CrossRef\]](#)
31. Koti, P.; Dhavachelvan, P.; KalaiPriyan, T.; Arjunan, S.; Uthayakumar, J.; Sujatha, P. An efficient healthcare framework for kidney disease using hybrid harmony search algorithm. *Electron. Gov. Int. J.* **2020**, *16*, 56–68. [\[CrossRef\]](#)
32. Degertekin, O. Optimum design of steel frames using harmony search algorithm. *Struct. Multidiscip. Optim.* **2008**, *36*, 393–401. [\[CrossRef\]](#)
33. Gao, Z.; Suganthan, P.N.; Pan, Q.K.; Chua, T.J.; Cai, T.X.; Chong, C.S. Discreet harmony search algorithm for flexible job shop scheduling problem with multiple objectives. *J. Intell. Manuf.* **2016**, *27*, 363–374. [\[CrossRef\]](#)
34. Tsakirakis, E.; Marinaki, M.; Marinakis, Y.; Matsatsinis, N. A similarity hybrid harmony search algorithm for the Team Orienteering problem. *Appl. Soft Comput.* **2019**, *80*, 776–796. [\[CrossRef\]](#)
35. Deng, W.; Xu, J.; Song, Y.; Zhao, H. Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. *Appl. Soft Comput.* **2021**, *100*, 106724. [\[CrossRef\]](#)
36. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.* **2021**, *25*, 5277–5298. [\[CrossRef\]](#)

37. Nabavi, M.; Elveny, M.; Danshina, S.D.; Behroyan, I.; Babanezhad, M. Velocity prediction of Cu/water nanofluid convective flow in a circular tube: Learning CFD data by differential evolution algorithm based fuzzy inference system (DEFIS). *Int. Commun. Heat Mass Transf.* **2021**, *126*, 105373. [[CrossRef](#)]
38. Houssein, E.H.; Mahdy, M.A.; Blondin, M.J.; Shebl, D.; Mohamed, W.M. Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Syst. Appl.* **2021**, *174*, 114689. [[CrossRef](#)]
39. Ahmadianfar, I.; Kheyrandish, A.; Jamei, M.; Gharabaghi, B. Optimizing operating rules for multi-reservoir hydropower generation systems. *Adapt. Hybrid Differ. Evol. Algorithm Renew. Energy* **2021**, *167*, 774–790.
40. Houssein, E.H.; Mahdy, M.A.; Eldin, M.G.; Shebl, D.; Mohamed, W.M.; Abdel-Aty, M. Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm. *J. Adv. Res.* **2021**, *29*, 147–157. [[CrossRef](#)]
41. Deng, W.; Liu, H.; Xu, J.; Zhao, H.; Song, Y. An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 7319–7327. [[CrossRef](#)]
42. Sallam, K.M.; Elsayed, S.M.; Chakraborty, R.K.; Ryan, M.J. Improved multi-operator differential evolution algorithm for solving unconstrained problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
43. Ochoa, P.; Castillo, O.; Soria, J. Optimization of fuzzy controller design using a differential evolution algorithm with dynamic parameter adaptation based on type-1 and interval type-2 fuzzy systems. *Soft Comput.* **2020**, *24*, 193–214. [[CrossRef](#)]
44. Guedria, N.B. An accelerated differential evolution algorithm with new operators for multi-damage detection in plate-like structures. *Appl. Math. Model.* **2020**, *80*, 366–383. [[CrossRef](#)]
45. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [[CrossRef](#)]
46. Coupland, S.; John, R. Geometric Type-1 and Type-2 Fuzzy Logic Systems. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 3–15. [[CrossRef](#)]
47. Wagner, C.; Hagrass, H. Toward General Type-2 Fuzzy Logic Systems Based on zSlices. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 637–660. [[CrossRef](#)]
48. Mendel, J.M.; Liu, F.; Zhai, D. Alpha-Plane Representation for Type-2 Fuzzy Sets: Theory and Applications. *IEEE Trans. Fuzzy Syst.* **2009**, *17*, 1189–1207. [[CrossRef](#)]
49. Mendel, J.M.; John, R.I.; Liu, F. Interval Type-2 Fuzzy Logic Systems Made Simple. *IEEE Trans. Fuzzy Syst.* **2006**, *14*, 808–821. [[CrossRef](#)]
50. Pedrycz, W. Shadowed sets: Representing and processing fuzzy sets. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1998**, *28*, 103–109. [[CrossRef](#)] [[PubMed](#)]
51. Pedrycz, W. From fuzzy sets to shadowed sets: Interpretation and computing. *Int. J. Intell. Syst.* **2009**, *24*, 48–61. [[CrossRef](#)]
52. Wijayasekara, D.; Linda, O.; Manic, M. Shadowed Type-2 Fuzzy Logic Systems. In *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*; IEEE: Piscataway, NJ, USA, 2013; pp. 15–22.
53. Zhang, T.; Geem, Z.W. Review of Harmony Search with Respect to Algorithm Structure. *Swarm Evol. Comput.* **2019**, *48*, 31–43. [[CrossRef](#)]
54. Tang, L.; Zhao, Y.; Liu, J. An Improved Differential Evolution Algorithm for Practical Dynamic Scheduling in Steelmaking-Continuous Casting Production. *IEEE Trans. Evol. Comput.* **2014**, *18*, 209–225. [[CrossRef](#)]
55. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
56. Ochoa, P.; Castillo, O.; Soria, J. Differential Evolution with Dynamic Adaptation of Parameters for the Optimization of Fuzzy Controllers. In *Recent Advances on Hybrid, Approaches for Designing Intelligent Systems*; Springer: Cham, Switzerland, 2014; pp. 275–288.
57. MATLAB. *Version 7.10.0 (R2013a)*; The MathWorks Inc.: Natick, MA, USA, 2013.