

Article

Comparative Study in Fuzzy Controller Optimization Using Bee Colony, Differential Evolution, and Harmony Search Algorithms

Oscar Castillo * , Fevrier Valdez, José Soria, Leticia Amador-Angulo, Patricia Ochoa and Cinthia Peraza

Tijuana Institute of Technology, c.p 22379 Tijuana, Mexico; fevrier@tectijuana.mx (F.V.); jsoria57@gmail.com (J.S.); gloria.amador@tectijuana.edu.mx (L.A.-A.); martha.ochoa18@tectijuana.edu.mx (P.O.); cinthia.peraza18@tectijuana.edu.mx (C.P.)

* Correspondence: ocastillo@tectijuana.mx; Tel.: +52-664-6236318

Received: 2 December 2018; Accepted: 23 December 2018; Published: 27 December 2018



Abstract: This paper presents a comparison among the bee colony optimization (BCO), differential evolution (DE), and harmony search (HS) algorithms. In addition, for each algorithm, a type-1 fuzzy logic system (T1FLS) for the dynamic modification of the main parameters is presented. The dynamic adjustment in the main parameters for each algorithm with the implementation of fuzzy systems aims at enhancing the performance of the corresponding algorithms. Each algorithm (modified and original versions) is analyzed and compared based on the optimal design of fuzzy systems for benchmark control problems, especially in fuzzy controller design. Simulation results provide evidence that the FDE algorithm outperforms the results of the FBCO and FHS algorithms in the optimization of fuzzy controllers. Statistically is demonstrated that the better errors are found with the implementation of the fuzzy systems to enhance each proposed algorithm.

Keywords: type-1 fuzzy logic; fuzzy controller; benchmark problems

1. Introduction

Nowadays meta-heuristic algorithms are used to solve various kinds of optimization problems, and this work is based on three particular algorithms, which are the BCO, DE, and HS algorithms.

In the last decade the BCO has proven to be an excellent technique in the optimization of nondeterministic polynomial time problems (NP-Problems) [1], like the following: fuzzy controllers [2–5], and general engineering problems [6–8].

The HS algorithm is inspired by the process of jazz improvisation, and various problems like the optimization of neural networks [9–11], benchmark functions [12–14], benchmark control [15], and engineering problems [16–19] have been successfully solved with this algorithm.

The DE algorithm belongs to the category of evolutionary computation. It efficiently solves nonlinear, non-differentiable and multimodal problems, and is used in the solution of complex problems [20]. There are also works that combine fuzzy logic with DE [21–23] and some control problem applications [24–26].

Many authors in the field of intelligent Systems are interested in control system stabilization [2,5,20,26]. In particular, for fuzzy systems in control is not necessary to have the exact mathematical models to achieve a good performance in control. It is because of this fact, that recently fuzzy control has demonstrated to be an excellent choice when applied for the stabilization in control problems. The first characteristic is the use the if-then rules which helps us in modeling knowledge [13,15,21,24,27]. An important aspect that this paper presents is adding an intelligent

system based on a meta-heuristic to improve the performance in the stabilization for the fuzzy controllers. In this case, the bee colony (BCO), differential evolution (DE), and harmony search (HS) algorithms are analyzed.

The following works demonstrate the importance of fuzzy logic and the use of metaheuristics to solve real problems of engineering, energy or management to mention some branches of application [28–32].

At present there are many metaheuristic algorithms, and these have been applied to solve real-world problems in different areas. In this paper the main contribution consists in a comparative study based on three bio-inspired algorithms for optimizing the design and implementation of fuzzy controllers, especially in benchmark control problems. In addition, the comparative study includes the traditional metaheuristic method, and the proposed methods with dynamic adjustment of parameters using type-1 fuzzy logic as a tool for modeling complex problems in control, in order to find which method is best for each type of problem.

This rest of the paper is organized as follows. Section 2 outlines the general concepts of the theory of fuzzy systems; Section 3 describes the benchmark control problems. Section 4 explains the methodology of the proposed methods. Section 5 presents the results of the simulations with all methodologies, Section 6 describes a statistical test. Section 7 presents a discussion of results, and finally in Section 8 the conclusions of the proposed method are offered.

2. Fuzzy Logic Systems

Zadeh in 1965 was the first one to introduce the main idea of fuzzy logic systems [33,34]. Equation (1) represents a fuzzy set in the universe U is characterized by a membership function $u_A(x)$ taking values on the interval $[0,1]$:

$$A = \{(x, u_A(x)) \mid x \in U\} \tag{1}$$

The main elements for a type-1 fuzzy logic system are defined in the following Figure 1.

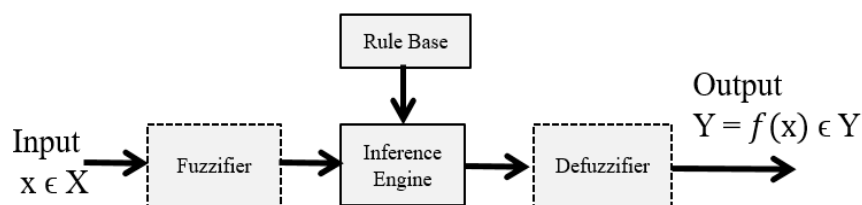


Figure 1. Architecture of a type-1 fuzzy logic system.

Figure 1 shows the main sections of a fuzzy logic system, which are; the fuzzifier that takes crisp inputs and maps them into fuzzy sets, the inference engine based on the rules, maps fuzzy sets from the antecedents to fuzzy sets from the consequents; finally, the defuzzifier obtains a crisp value from the fuzzy sets.

An example of the rules is represented by Equation (2), and input space is mapped to the output space.

$$R^l : \text{IF } x_1 \text{ is } F_1^l \text{ and } \dots \text{ and } x_p \text{ is } F_p^l, \text{ THEN } y \text{ is } G^l, \text{ where } l = 1, \dots, M \tag{2}$$

where R^l indicates a specific rule, x_p is input p , F_p^l is a membership function on rule l and input p , y is the output on membership function G^l . Both F and G are in the form of $\mu_F(x)$ and $\mu_G(y)$, respectively.

3. Benchmark Problems

In the literature there are several models of fuzzy control systems, one of the most used is the Mamdani model [35,36] which operates with the modules described in Figure 1, the other is the Takagi-Sugeno-Kang (TSK) model [37,38], which is an alternative model in which the consequent does

not give us a fuzzy set but a linear function. The main difference between the TSK method and the Mamdani method is that in TSK is not necessary to perform a defuzzifier process.

To validate the proposed algorithms in this work, two benchmark control problems were selected, which are the water tank that uses a Mamdani model and the inverted pendulum that uses a TSK model.

3.1. Water Tank Control Problem

The first benchmark control problem is the water level control in a tank [39,40] and is illustrated in Figure 2.

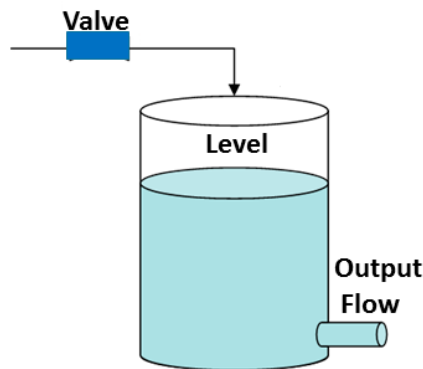


Figure 2. Water tank system.

The mathematical model of this controller is expressed in Equation (3):

$$h(t) = h(0) + \int_0^t \frac{1}{A} (q_{in}(t') - q_{out}(t')) dt' = \int_0^t \frac{1}{A} \left(q_{in}(t') - a\sqrt{2gh(t')} \right) dt' \quad (3)$$

The fuzzy system for this problem is of Mamdani type, used the max aggregation and the centroid defuzzification, which is composed of two inputs which have three triangular membership functions and a one output which has five triangular membership functions, as represented in Figure 3, the fuzzy rules that are used are shown in Figure 4.

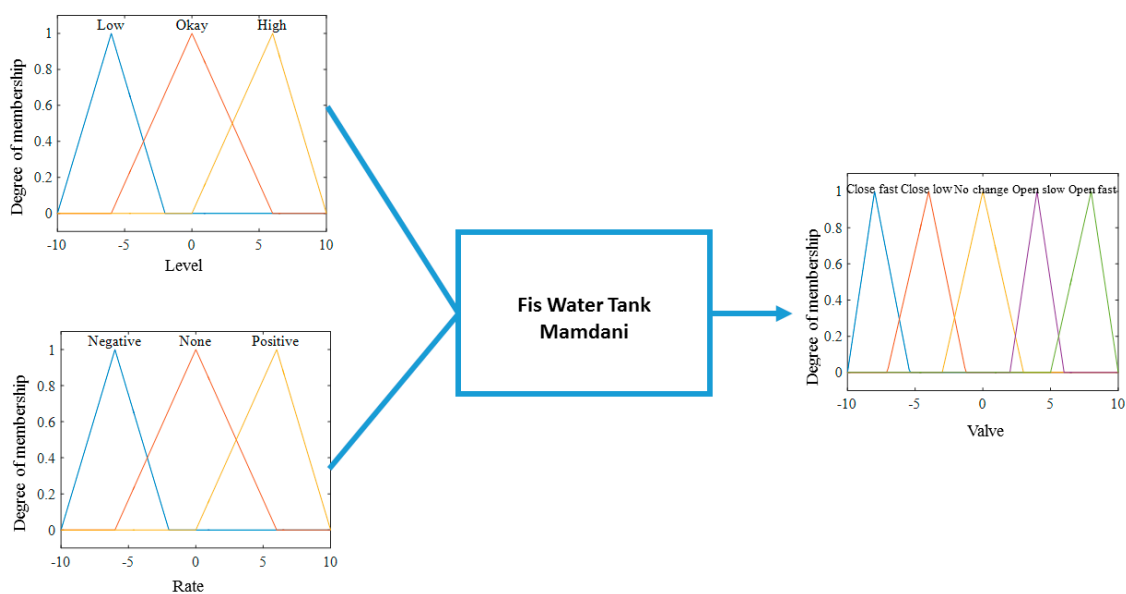


Figure 3. Water tank system.

1. If (level is Okay) then (valve is no_change) (1)
2. If (level is High) then (valve is open_fast) (1)
3. If (level is Low) then (valve is close_fast) (1)
4. If (level is Okay) and (rate is Positive) then (valve is close_slow) (1)
5. If (level is Okay) and (rate is Negative) then (valve is open_slow) (1)

Figure 4. Fuzzy rules for the water tank controller.

The fuzzy rules that are outlined in Figure 4 were obtained based on analyzing the filling of a water tank [5]. Basically the knowledge needed to control the valve in filling the tank is used for designing the rules.

3.2. Control of an Inverted Pendulum on a Cart

The main feature of this controller is a complex non-linear control problem, with its complexity originating from the nonlinear nature of the plant [41,42]. The main goal of the controller is to apply a force to move the cart so that the pendulum remains in the vertical unstable position. The cart pole system is shown in Figure 5, where x is the cart position, θ is the pendulum angle, and F is the control force applied, parallel to the rail, to the cart, where $g = 9.81 \text{ m/s}^2$.

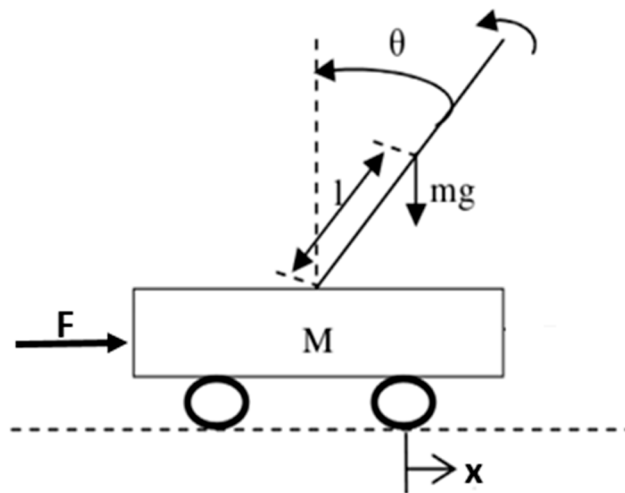


Figure 5. Inverted pendulum system.

The fuzzy system for this problem is of Takagi-Sugeno type, using the max aggregation and the weighted average defuzzification, which is composed of four inputs with triangular membership functions and a one output with 16 linear functions, as represented in Figure 6, and the fuzzy rules that are considered are shown in Figure 7.

The fuzzy rules that are illustrated in Figure 7 are those that maintain control of the cart for the inverted pendulum.

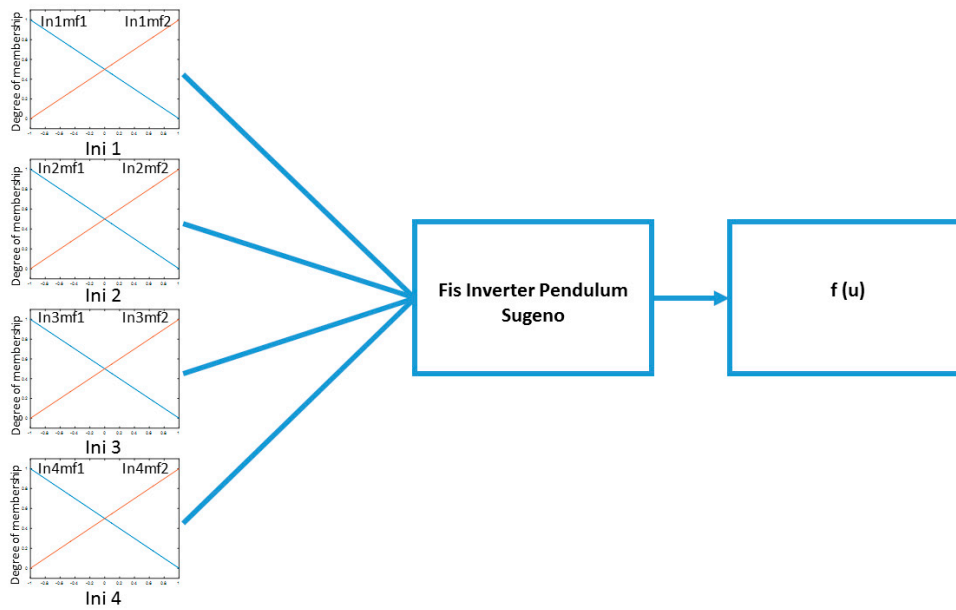


Figure 6. Inverted pendulum system.

1. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf1) (1)
2. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf2) (1)
3. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf3) (1)
4. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf4) (1)
5. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf5) (1)
6. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf6) (1)
7. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf7) (1)
8. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf8) (1)
9. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf9) (1)
10. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf10) (1)
11. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf11) (1)
12. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf12) (1)
13. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf13) (1)
14. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf14) (1)
15. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf15) (1)
16. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf16) (1)

Figure 7. Fuzzy rules for the inverted pendulum.

4. Proposed Fuzzy Algorithms

A T1FLS is designed as a strategy to dynamically adjust the parameters of each algorithm and in this way aiming at controlling the exploitation and exploration abilities of the algorithms in the search space.

For each of the algorithms a fuzzy system is used in which the inputs are formed by iterations or generations depending on the method, this can be observed in the Equation (4), and a second input that is the diversity, which is the Euclidean distance among the possible solutions, which helps depending on the algorithm to achieve exploitation and exploration, and is defined by Equation (5).

$$Iteration = \frac{Current\ iteration}{Maximum\ of\ iterations} \tag{4}$$

$$Diversity\ (S(t)) = \frac{1}{n_S} \sum_{i=1}^{n_S} \sqrt{\sum_{j=1}^{n_x} (x_{ij}(t) - \bar{x}_j(t))^2} \tag{5}$$

4.1. Bee Colony Optimization Algorithm

The methodology of the original BCO algorithm and fuzzy bee colony optimization algorithm (FBCO) are described in this section.

4.1.1. Original Bee Colony Optimization Algorithm (BCO)

The behavior of a bee in nature is the main inspiration for the BCO algorithm. This methodology is based on creation of multi agent system (colony of artificial bees) capabilities to successfully solve difficult combinatorial optimization problems [43–45]. BCO was proposed by Teodorović in 2001. The dynamics in BCO are represented by the following equations:

$$P_{ij,n} = \frac{[\rho_{ij,n}]^\alpha \cdot [\frac{1}{d_{ij}}]^\beta}{\sum_{j \in A_{i,n}} [\rho_{ij,n}]^\alpha [\frac{1}{d_{ij}}]^\beta} \tag{6}$$

$$D_i = K \cdot \frac{P_{f_i}}{P_{f_{colony}}} \tag{7}$$

$$P_{f_i} = \frac{1}{L_i}, L_i = \text{Tour Length} \tag{8}$$

$$P_{f_{colony}} = \frac{1}{N_{Bee}} \sum_{i=1}^{N_{Bee}} P_{f_i} \tag{9}$$

4.1.2. Fuzzy Bee Colony Optimization Algorithm (FBCO)

The FBCO algorithm is characterized by its dynamic parameters different to the original bee colony optimization algorithm. This fuzzy system is of Mamdani type and using the max aggregation and the centroid defuzzification. The parameters used are illustrated in Figure 8, the input parameters are based on Equations (4) and (5), and the output ones are specific to the algorithm, which are the beta and alpha parameters. In BCO beta represents the exploration and alpha exploitation, respectively. Both parameters can be denoted by Equation (6).

The selection of rules in the fuzzy system is based on the behavior of the algorithm parameters to achieve the control of the exploration in early iterations and the exploitation in final iterations, which are shown in Table 1.

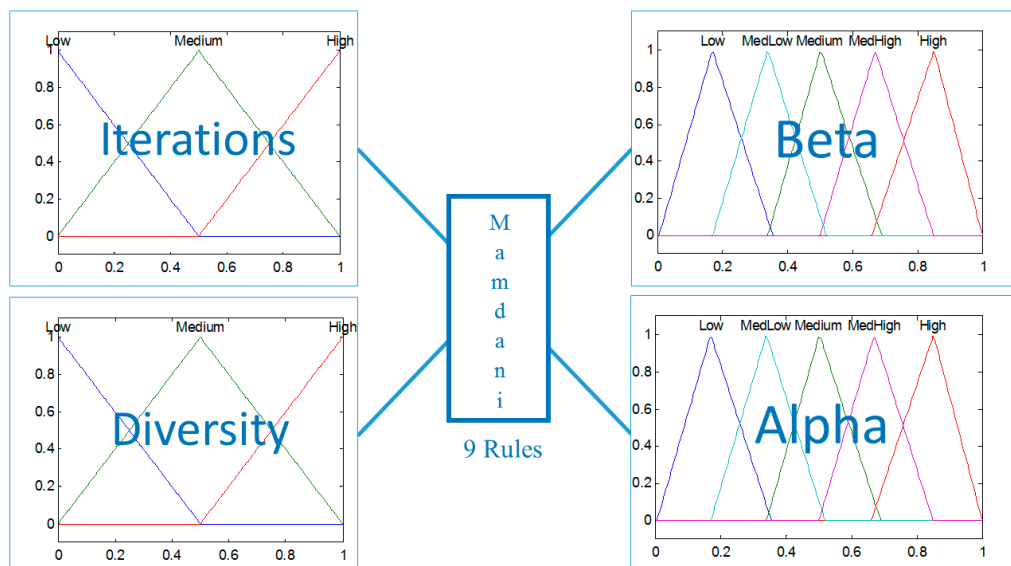


Figure 8. Fuzzy bee colony algorithm.

Table 1. Fuzzy rules for the FBCO.

| Iterations | Inputs | | | Output | |
|------------|----------|-----------|------|-------------|-------------|
| | Operator | Diversity | | Beta | Alpha |
| Low | and | Low | then | High | Low |
| Low | and | Medium | then | Medium High | Medium |
| Low | and | High | then | Medium High | Medium Low |
| Medium | and | Low | then | Medium High | Medium Low |
| Medium | and | Medium | then | Medium | Medium |
| Medium | and | Low | then | Medium | Medium High |
| High | and | Low | then | Medium Low | High |
| High | and | Medium | then | Medium Low | Medium High |
| High | and | High | then | Low | High |

Figure 9 illustrates the pseudocode of the original BCO method.

```

Generate the Initial population of individuals
Determine initial solutions
Evaluate the initial solutions
S ← the best solution of the bees
while (t<Maximum number of iterations)
  Do
    For each iteration bee in the population
      Do
        For each bee follower in the population
          Set an initial solution
          Evaluate modified solutions generated by possible changes
          By roulette wheel choose one the modified solutions
          Evaluate news solutions
          Made a decision whether the best is royal
          If the bee is not royal then
            Choice one of the royal bees followed by the it-h bee
          End If
        End for
      End for
      If the best solution of the bees better the solution S then
        S ← the best bee's solution
      End if
      t = t + 1
    End while
  To find the best solution (S)
  
```

Figure 9. Bee colony optimization.

Figure 10 illustrates the pseudocode of the modified BCO with parameter adaptation. The main difference with respect to Figure 9 is can be found in calculate the iteration and diversity values and to find the new beta and alpha parameters through of the fuzzy logic system.

4.2. Differential Evolution Algorithm

This section describes the methodology of the original DE algorithm and the fuzzy differential evolution algorithm (FDE) [21,22].

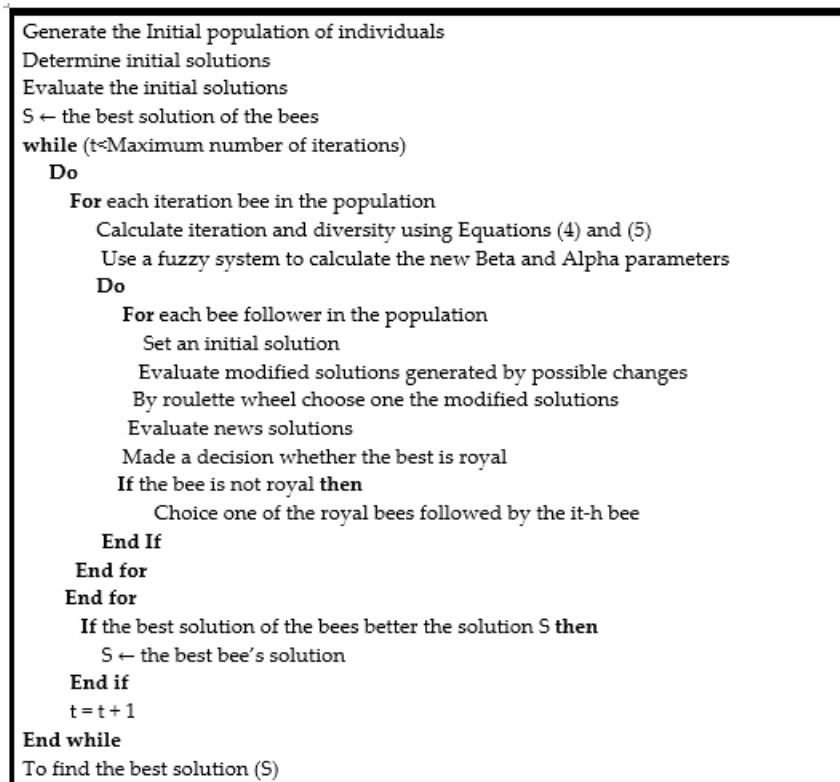


Figure 10. Fuzzy bee colony optimization.

4.2.1. Differential Evolution Algorithm

DE is one of the most used algorithms; it is also one of the simplest that exist in the literature. This is the mathematical representation of the DE:

Population structure:

$$P_{x,g} = (x_{i,g}), i = 0, 1, \dots, Np - 1, g = 0, 1, \dots, g_{max}, x_{i,g} = (x_{j,i,g}), j = 0, 1, \dots, D - 1 \quad (10)$$

$$P_{v,g} = (v_{i,g}), i = 0, 1, \dots, Np - 1, g = 0, 1, \dots, g_{max}, v_{i,g} = (v_{j,i,g}), j = 0, 1, \dots, D - 1 \quad (11)$$

$$P_{u,g} = (u_{i,g}), i = 0, 1, \dots, Np - 1, g = 0, 1, \dots, g_{max}, u_{i,g} = (u_{j,i,g}), j = 0, 1, \dots, D - 1 \quad (12)$$

Initialization:

$$x_{j,i,0} = rand_j(0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L} \quad (13)$$

Mutation:

$$v_{i,g} = x_{r_0,g} + F \cdot (x_{r_1,g} - x_{r_2,g}) \quad (14)$$

Crossover:

$$u_{i,g} = u_{j,i,g} \begin{cases} v_{j,i,g} & \text{if } (rand_j(0,1) \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (15)$$

Selection:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases} \quad (16)$$

4.2.2. Fuzzy Differential Evolution Algorithm (FDE)

The FDE algorithm is characterized by its dynamic parameters different to the original DE algorithm. This Fuzzy system is Mamdani type and used the max aggregation and the centroid

defuzzification. The parameters used are illustrated in Figure 11, the input parameters are based on Equations (4) and (5), and the output ones are specific to the algorithm, which are the mutation rate (F) and crossover rate (CR) parameters.

The rules of this fuzzy system are based on the behavior of the algorithm parameters to achieve the control of the exploration in early iterations and the exploitation in final iterations, which are shown in Table 2.

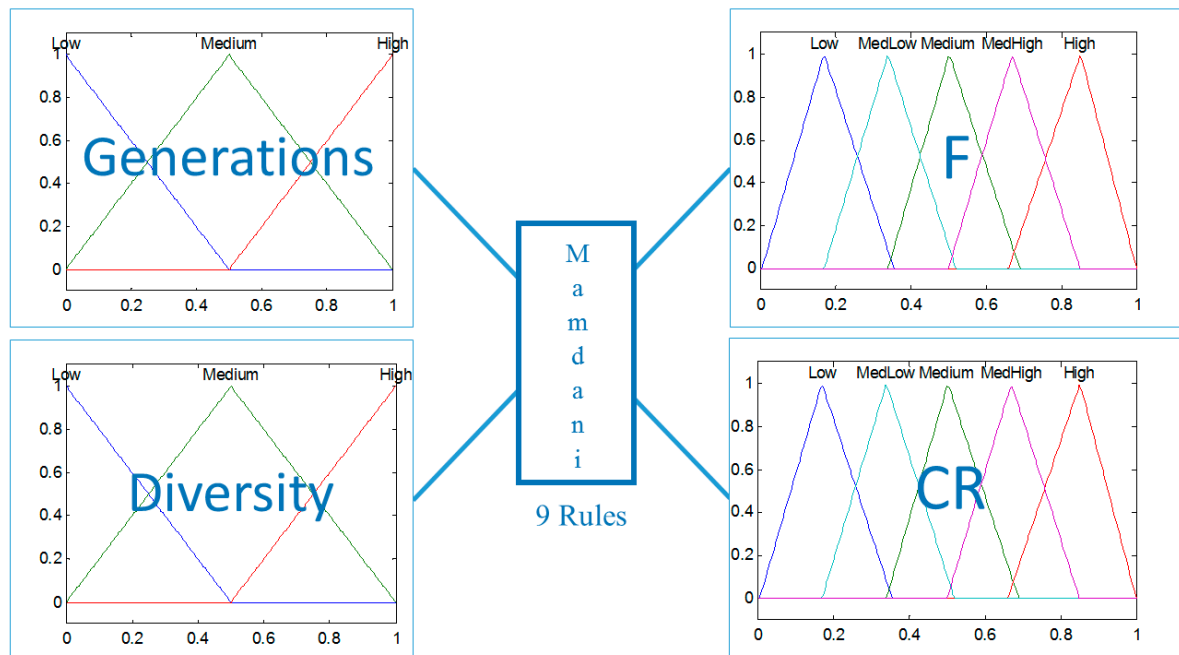


Figure 11. Fuzzy differential evolution algorithm.

Table 2. Fuzzy rules for the FDE.

| Inputs | | | Output | | |
|-------------|----------|-----------|--------|-------------|-------------|
| Generations | Operator | Diversity | | F | CR |
| Low | and | Low | then | High | Low |
| Low | and | Medium | then | High | Low |
| Low | and | High | then | Medium high | Medium Low |
| Medium | and | Low | then | Medium High | Medium Low |
| Medium | and | Medium | then | Medium | Medium |
| Medium | and | High | then | Medium | Medium |
| High | and | Low | then | Medium Low | Medium high |
| High | and | Medium | then | Medium Low | Medium High |
| High | and | High | then | Low | High |

Figure 12 illustrates the pseudocode of the original DE method.

Figure 13 illustrates the pseudocode of the modified FDE with parameter adaptation. The main difference with respect to Figure 12 can be found in calculate the generation and diversity values and to find the new mutation and crossover parameters through of the fuzzy logic system.

4.3. Harmony Search Algorithm

This section describes the methodologies of the original HS algorithm and fuzzy harmony search algorithm (FHS).

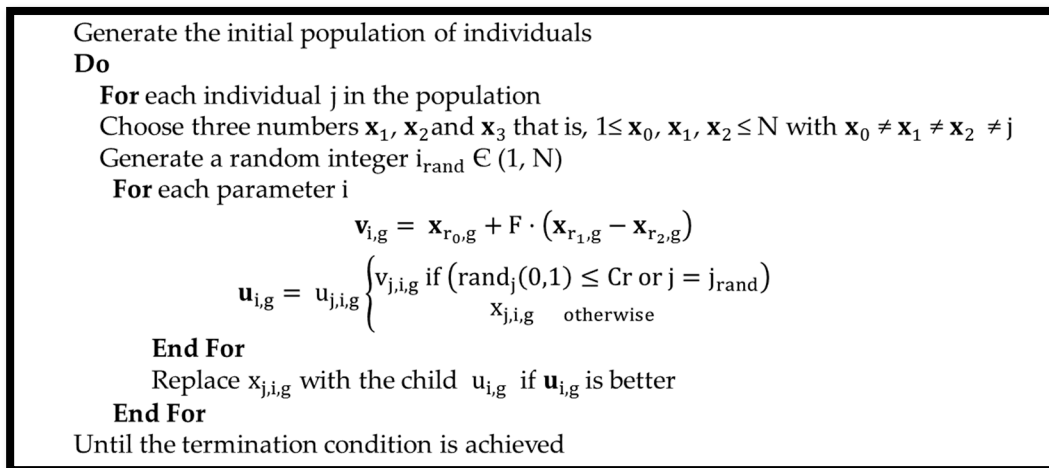


Figure 12. Differential evolution algorithm.

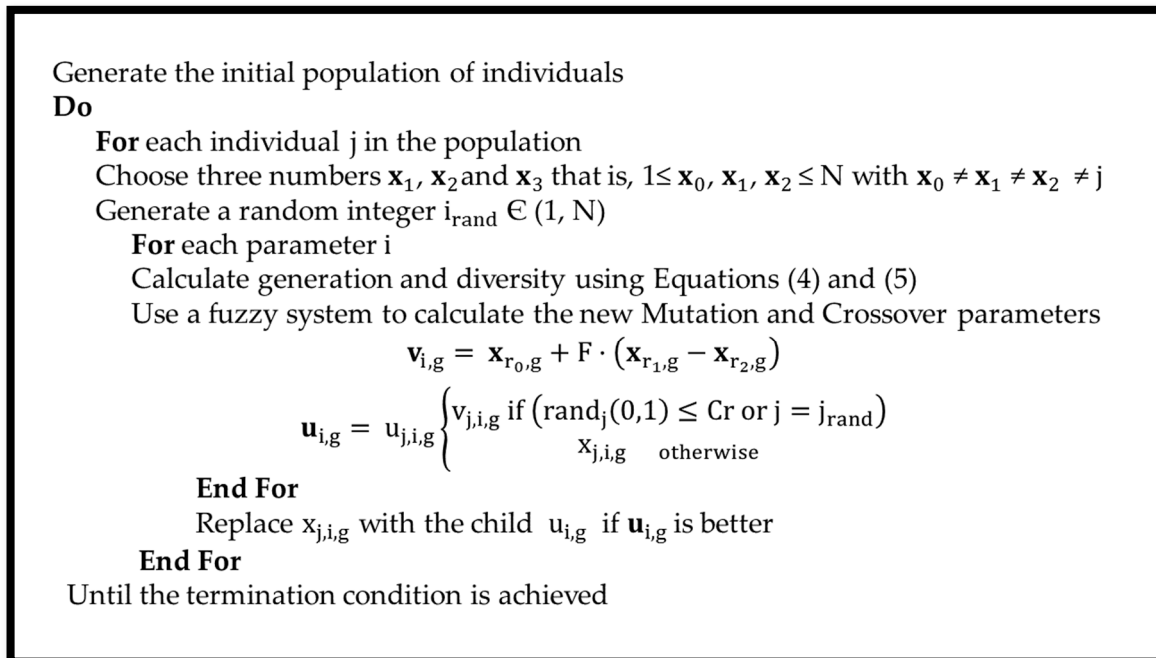


Figure 13. Fuzzy differential evolution algorithm.

4.3.1. Original Harmony Search Algorithm (HS)

Zong Woo Geem was the first to create this algorithm in 2001, and it is inspired by the improvisation of music, specifically referred to jazz [46]. The most important elements that make up this algorithm are: harmony memory accepting (HMR $\in [0,1]$) this parameter represents the exploitation of the search space, random selection and pitch adjustment (PARate $\in [0,1]$) and these parameters represent the exploration of the search space and are expressed by the following Equations [47]:

$$x_{new} = x_{old} + b_p(2 \text{ rand} - 1) \tag{17}$$

$$P_{Arate} = p_{lower \ limit} + P_{range} * \text{rand} \tag{18}$$

$$P_{range} = P_{upper \ limit} - P_{lower \ limit} \tag{19}$$

$$P_{random} = 1 - HMR \tag{20}$$

$$P_{pitch} = HMR * P_{Arate} \tag{21}$$

where x_{old} is the current solution, x_{new} is the new solution, and $rand$ is a generator of random numbers in the range of 0 to 1 [48].

4.3.2. Fuzzy Harmony Search Algorithm (FHS)

The main difference between the HS and the FHS algorithm is the dynamic parameter adaptation of the algorithm using a fuzzy system based on rules created based on the behavior of the algorithm parameters to achieve the control of the exploration in early iterations and the exploitation in final iterations, which are shown in Table 3. This fuzzy system is Mamdani type and used the max aggregation and the centroid defuzzification.

Specifically, the HMR and PARate parameters are adapted, as shown in Figure 14. The metrics for the input parameters of the fuzzy system are “iterations” and “diversity”, which are based on Equations (4) and (5):

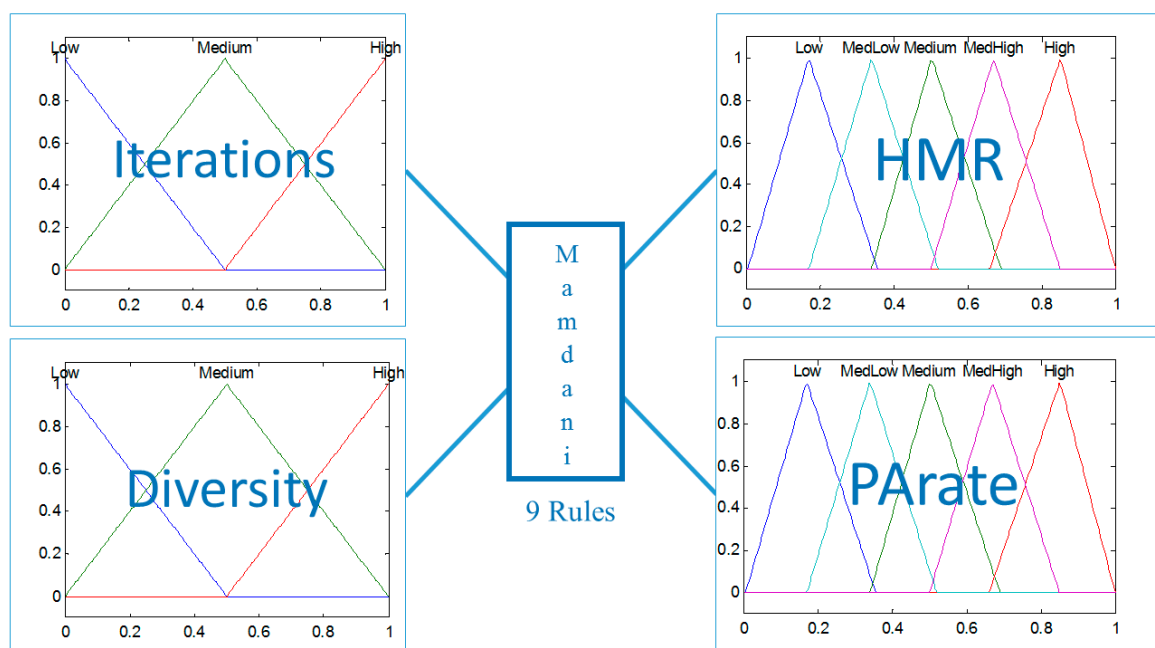


Figure 14. Fuzzy harmony search algorithm.

Table 3. Fuzzy rules for the FHS.

| Inputs | | | Output | | |
|-------------|----------|-----------|--------|-------------|-------------|
| Generations | Operator | Diversity | | HMR | PARate |
| Low | and | Low | then | High | Low |
| Low | and | Medium | then | Medium Low | Medium High |
| Low | and | High | then | Medium | Medium |
| Medium | and | Low | then | Medium | Medium Low |
| Medium | and | Medium | then | Medium | Medium |
| Medium | and | High | then | Medium | High |
| High | and | Low | then | Medium | High |
| High | and | Medium | then | Medium High | Medium Low |
| High | and | High | then | High | High |

Figure 15 illustrates the pseudocode of the original HS method.

Figure 16 illustrates the pseudocode of the modified FHS with parameter adaptation. The main difference with respect to Figure 15 can be found in calculate the iteration and diversity values and to find the new HMR and PARate parameters through of the fuzzy logic system.

```

Define objective function  $f(x), x = (X_1, \dots, X_n)^T$ 
Initial generate harmonics (matrix of real numbers)
Define Pitch adjustment rate (PARate) and limits of tone
Define harmony memory accepting (HMR)
while (t<Maximum number of iterations)
    Generate a new harmony and accept the best system
    Setting the tone for new harmonies (solutions)
        If (rand > HMR)
            Choose an existing harmony randomly
        else if (rand > Parate)
            Setting the tone at random within a bandwidth
        else
            Generate a new harmony through a randomization
        end if
    Accepting new harmonies (solutions) best
end while
To find the best solutions

```

Figure 15. Harmony search algorithm.

```

Define objective function  $f(x), x = (X_1, \dots, X_n)^T$ 
Initial generate harmonics (matrix of real numbers)
Define Pitch adjustment rate (PARate) and limits of tone
Define harmony memory accepting (HMR)
while (t<Maximum number of iterations)
    Calculate iteration and diversity using Equations (4) and (5)
    Calculate new HMR parameter using a fuzzy system
    Generate a new harmony and accept the best system
    Setting the tone for new harmonies (solutions)
        if (rand > HMR)
            Choose an existing harmony randomly
        else if (rand > Parate)
            Setting the tone at random within a bandwidth
        else
            Generate a new harmony through a randomization
        end if
    Accepting new harmonies (solutions) best
end while
To find the best solutions

```

Figure 16. Fuzzy harmony search algorithm.

5. Simulation Results

Previously, the two study cases of the Water tank and Inverted pendulum were presented, which are used to validate the three proposed methods FBCO, FDE, and FHS. Table 4 shows the parameters that were used in each original method and Table 5 shows the parameters that were used in each of the proposed methods, respectively.

Table 4. Parameters used with the original methods.

| BCO | | DE | | HS | |
|----------------|------|----------------|-----|------------|------|
| Iterations | 100 | Generations | 100 | Iterations | 100 |
| NP(Population) | 50 | NP(Population) | 50 | Harmonies | 50 |
| FoodNumber | NP/2 | F (mutation) | 0.7 | HMR | 0.95 |
| Alpha | 0.5 | CR (crossover) | 0.3 | PARate | 0.70 |
| Beta | 2.5 | - | - | - | - |

Table 5. Parameters used with the proposed methods.

| FBCO | | FDE | | FHS | |
|----------------|---------|----------------|---------|------------|---------|
| Iterations | 100 | Generations | 100 | Iterations | 100 |
| NP(Population) | 50 | NP(Population) | 50 | Harmonies | 50 |
| FoodNumber | NP/2 | F (mutation) | Dynamic | HMR | Dynamic |
| Alpha | Dynamic | CR (cruce) | Dynamic | PARate | Dynamic |
| Beta | Dynamic | - | - | - | - |

The main goal of the experiments carried out by each of the algorithms is minimizing RMSE, which represents the root mean square error and it is defined in Equation (22):

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \tag{22}$$

The results obtained are presented in Sections 5.1–5.3 which include the best results, the worst, the averages of 30 experiments and the standard deviations of the 30 experiments.

The comparison presented in the tables correspond to the simulation of each control problem with the original algorithm, the algorithm with the dynamic adaptation of parameters and finally the problem of control applying noise of a uniform random number generator using the algorithm with dynamic adaptation.

5.1. Simulations Results for the FBCO

Experimentation was performed with external perturbations, which means that the uniform random number was added. The setting for noise was a sample time of 0.1 and the range of −0.05 to 0.05. Table 6 represents the results obtained for the water tank controller using FBCO algorithm and this Table 7 shows the results for the inverted pendulum on a cart controller.

Table 6. Results obtained from the water tank controller by BCO and FBCO algorithms.

| Water Tank Controller | | | | |
|-----------------------|--------------------------|-----------------------|---------------------------|------------------------|
| RMSE | BCO FLC without Noise | BCO FLC with Noise | FBCO FLC without Noise | FBCO FLC with Noise |
| Best | 4.50×10^{-1} | 3.64×10^{-1} | 4.24×10^{-1} | 3.64×10^{-1} |
| Worst | 5.91×10^{-1} | 5.06×10^{-1} | 5.69×10^{-1} | 5.06×10^{-1} |
| Average | 5.21×10^{-1} | 4.50×10^{-1} | 4.96×10^{-1} | 2.98×10^{-2} |
| Standard Deviation | 3.47×10^{-1} | 2.98×10^{-2} | 3.81×10^{-1} | 4.50×10^{-1} |

The results obtained for each one of the controllers show that the use of the proposed methodology works for the BCO algorithm, in both controllers we can see that when applying to the plant noise the result is still stable and for the case of the water tank the result on average is better, but not for the inverted pendulum controller.

Table 7. Results obtained from the inverted pendulum controller by BCO and FBCO algorithms.

| Inverted Pendulum on a Cart | | | | |
|-----------------------------|--------------------------|-----------------------|---------------------------|------------------------|
| RMSE | BCO FLC without Noise | BCO FLC with Noise | FBCO FLC without Noise | FBCO FLC with Noise |
| Best | 4.37×10^{-1} | 3.78×10^{-1} | 3.79×10^{-1} | 3.80×10^{-1} |
| Worst | 5.61×10^{-1} | 8.64×10^{-1} | 8.77×10^{-1} | 9.47×10^{-1} |
| Average | 5.07×10^{-2} | 6.02×10^{-1} | 8.21×10^{-2} | 4.82×10^{-1} |
| Standard Deviation | 3.23×10^{-1} | 3.45×10^{-1} | 4.66×10^{-1} | 9.19×10^{-2} |

5.2. Simulations Results for the FDE

The results of the DE and FDE are shown in Tables 8 and 9, for the water tank controller and inverted pendulum on a cart controller, respectively.

Table 8. Results obtained from the water tank controller by DE and FDE algorithm.

| Water Tank Controller | | | | |
|-----------------------|-------------------------|-----------------------|--------------------------|-----------------------|
| RMSE | De FLC Without Noise | DE FLC With Noise | FDE FLC Without Noise | FDE FLC with Noise |
| Best | 1.65×10^{-2} | 1.08×10^{-1} | 4.82×10^{-2} | 1.27×10^{-3} |
| Worst | 2.36×10^{-1} | 4.87×10^{-1} | 1.82×10^{-1} | 2.32×10^{-1} |
| Average | 9.37×10^{-2} | 4.48×10^{-1} | 9.44×10^{-2} | 3.67×10^{-2} |
| Standard Deviation | 4.81×10^{-2} | 7.43×10^{-2} | 3.34×10^{-2} | 5.60×10^{-2} |

Table 9. Results obtained from the inverted pendulum controller by DE and FDE algorithms.

| Inverted Pendulum on a Cart | | | | |
|-----------------------------|-------------------------|-----------------------|--------------------------|-----------------------|
| RMSE | DE FLC without Noise | DE FLC with Noise | FDE FLC without Noise | FDE FLC with Noise |
| Best | 8.45×10^{-2} | 2.96×10^{-1} | 1.83×10^{-2} | 1.40×10^{-2} |
| Worst | 1.46×10^0 | 2.99×10^{-1} | 1.40×10^0 | 7.92×10^{-1} |
| Average | 8.99×10^{-1} | 2.97×10^{-1} | 2.15×10^{-1} | 3.78×10^{-1} |
| Standard Deviation | 4.15×10^{-1} | 6.28×10^{-4} | 2.75×10^{-1} | 1.90×10^{-1} |

In the case of the DE algorithm, the results for the water tank controller improve according to the complexity, we refer to the case when the controller has noised the response of the proposal used is good on average and compared to the best result in the different variants.

For the case of the inverted pendulum we can say that we have improvement compared to the original algorithm but with the controller with noise added the results obtained are competitive but not better.

5.3. Simulations Results for the FHS

Table 10 summarizes the results obtained for the water tank controller using FHS algorithm and Table 11 shows the results for the inverted pendulum on a cart controller.

Table 10. Results obtained from the water tank controller by HS and FHS algorithms.

| Water Tank Controller | | | | |
|-----------------------|-------------------------|-----------------------|--------------------------|-----------------------|
| RMSE | HS FLC without Noise | HS FLC with Noise | FHS FLC without Noise | FHS FLC with Noise |
| Best | 4.98×10^{-1} | 1.01×10^{-2} | 4.86×10^{-1} | 2.53×10^{-2} |
| Worst | 6.24×10^{-1} | 6.99×10^{-1} | 6.25×10^{-1} | 2.01×10^{-1} |
| Average | 5.66×10^{-1} | 1.40×10^{-1} | 5.54×10^{-1} | 8.36×10^{-2} |
| Standard Deviation | 3.26×10^{-2} | 1.68×10^{-1} | 2.92×10^{-2} | 3.52×10^{-2} |

Table 11. Results obtained from the inverted pendulum controller by HS and FHS algorithms.

| Inverted Pendulum on a Cart | | | | |
|-----------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| RMSE | HS FLC without Noise | HS FLC with Noise | FHS FLC without Noise | FHS FLC with Noise |
| Best | 3.15×10^{-1} | 3.97×10^{-1} | 2.99×10^{-1} | 2.97×10^{-1} |
| Worst | 4.88×10^0 | 1.28×10^0 | 2.01×10^0 | 1.76×10^{-1} |
| Average | 1.88×10^0 | 1.02×10^0 | 7.67×10^{-1} | 7.54×10^{-1} |
| Standard Deviation | 1.35×10^0 | 4.07×10^{-1} | 4.81×10^{-1} | 4.49×10^{-1} |

Finally, the results obtained for the HS algorithm, in both controllers we can see that there is an improvement compared to the original algorithm, for the water tank problem the average difference is as we expected since the noise controller has better results than the other cases, and in the case of the inverted pendulum four is a minimum difference in average between the three cases.

A comparative of the RMSE values with the original and the fuzzy algorithms without noise is illustrated in Figure 17 for the water tank controller, and Figure 18 shows the comparison for the inverted pendulum on a cart controller.

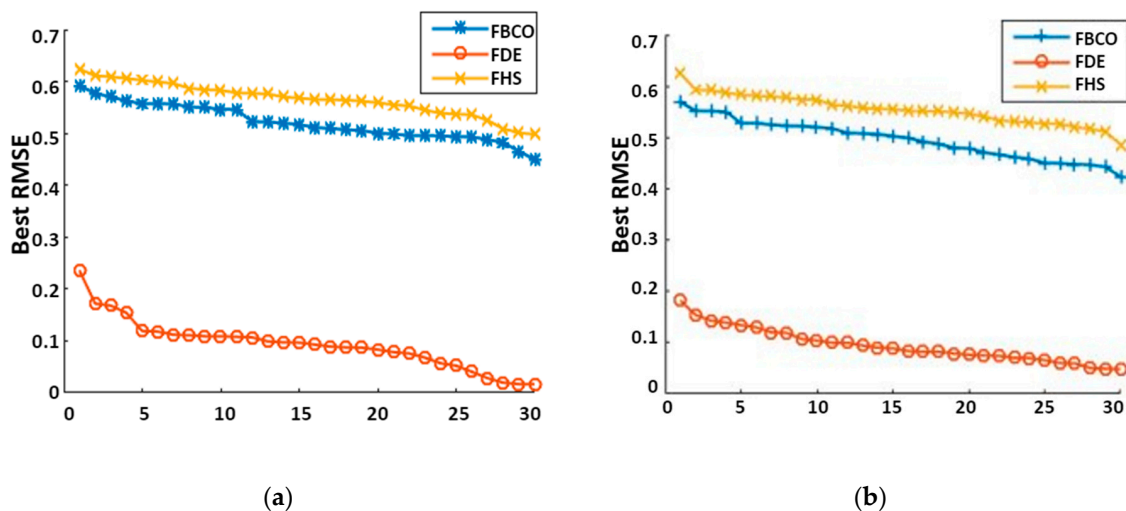


Figure 17. Comparison of the RMSE for each method for the water tank controller: (a) original algorithms, (b) fuzzy algorithm.

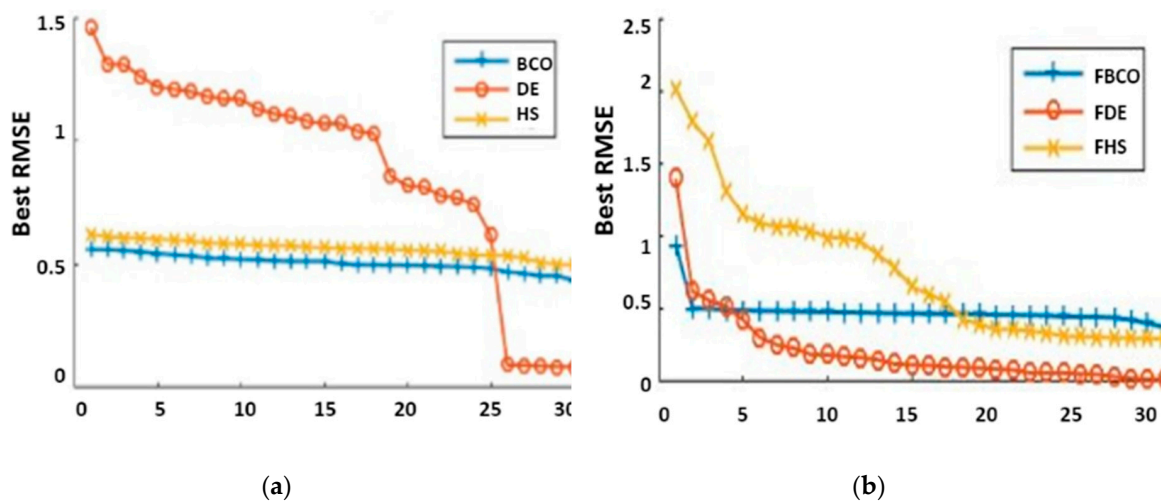


Figure 18. Comparison of the RMSE for each method for the inverted pendulum on a cart controller: (a) original algorithms, and (b) fuzzy algorithm.

Figure 17 shows that the DE found the better results compared to BCO and HS with the original and fuzzy algorithms for the water tank controller.

Figure 18 shows that DE found the better results compared to BCO and HS with the original and fuzzy algorithms for the inverted pendulum on a cart controller.

With the goal to analyze the behavior of each algorithm, Figure 19 shows a comparison of each fuzzy algorithm with noise in the model for the two benchmark problems.

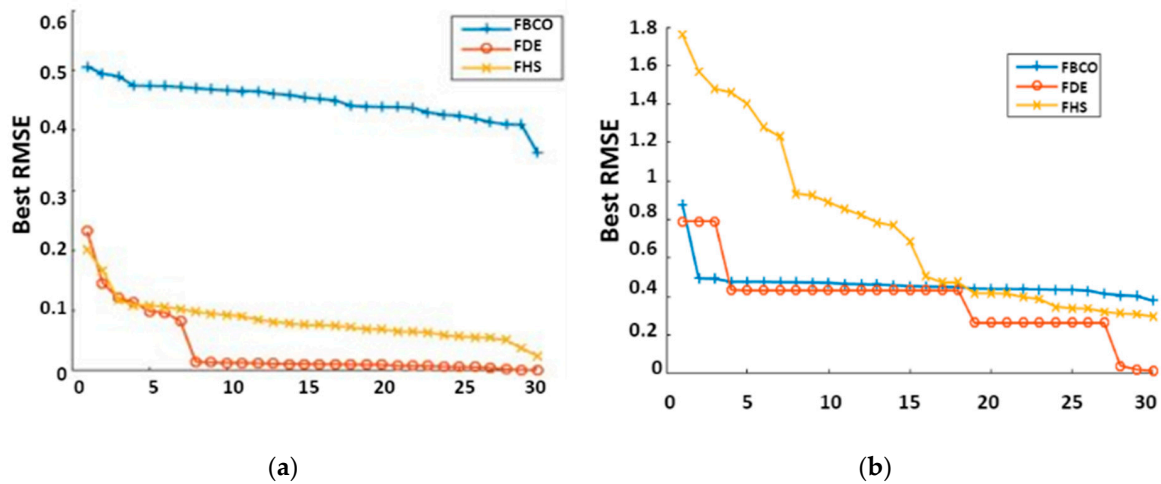


Figure 19. Comparison of the RMSE for each method for both benchmark problems with noise: (a) water tank controller, and (b) inverted pendulum on a cart controller.

Figure 19 shows that DE algorithm is better when noise is added in the model compared to BCO and HS in both benchmark problems.

Tables 12 and 13 show the times obtained when using the three original methods with noise and without noise (BCO, DE, HS) and the three proposed methods with noise and without noise (FBCO, FDE, FHS) for each FLC (water tank and inverted pendulum). It should be noted that the time shown is in seconds that lasted for one experiment.

Table 12. Comparison of the time obtained from the water tank controller for the original and proposed algorithms.

| Water Tank Controller | | | | |
|-----------------------|----------------------------|-------------------------|----------------------------|-------------------------|
| Method | Original FLC without Noise | Original FLC with Noise | Proposed FLC without Noise | Proposed FLC with Noise |
| BCO | 381.603 | 515.283 | 408.816 | 495.371 |
| DE | 205.357 | 237.064 | 228.372 | 268.867 |
| HS | 890.95 | 65.63 | 885.70 | 77.16 |

Table 13. Comparison of the time obtained from the inverted pendulum controller for the original and proposed algorithms.

| Inverted Pendulum on a Cart | | | | |
|-----------------------------|------------------------|---------------------|------------------------|---------------------|
| Method | Original without Noise | Original with Noise | Proposed without Noise | Proposed with Noise |
| BCO | 323.84 | 298.92 | 492.03 | 682.54 |
| DE | 310.66 | 380.76 | 474.44 | 682.14 |
| HS | 316.08 | 2326.63 | 508.74 | 1173.25 |

Times vary depending on the complexity of the FLC but the fuzzy method takes a little longer but in most cases, it is more efficient.

6. Statistical Test

There are parametric and nonparametric tests to test a statistical hypothesis. The parametric test indicates which sample is better, the non-parametric ones indicate if there is a difference in the samples. In this case, a parametric statistical test is chosen to test which method is the best one for each of the problems presented above.

To complete the study carried out in this paper, a statistical test is presented in this section to compare the three fuzzy algorithms FBCO, FDE, and FHS, with the two control problems applied in the previous sections. The selected statistical test is Z test and the parameters used are described in Table 14.

Table 14. Parameters for the statistical z-test.

| Parameter | Value |
|---------------------|--------------------|
| Level of Confidence | 95% |
| Alpha | 0.05% |
| H _a | $\mu_1 < \mu_2$ |
| H ₀ | $\mu_1 \geq \mu_2$ |
| Critical Value | -1.645 |

The hypotheses used for the statistical test indicate the following:

H₀: The proposed Type-1 algorithm without noise and with noise is greater than or equal to the originals algorithm.

H_a: The proposed Type-1 algorithm without noise and with noise is smaller than the originals algorithm.

According to the values used in Table 14, there is a rejection zone for values lower than -1.64. The Equation (23) of the z-test is shown as follows:

$$Z = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sigma_{\bar{X}_1 - \bar{X}_2}} \tag{23}$$

Tables 15–17 show the values of Z, and in these tables “S” means that is found evidence of significance and “N.S” refers to which no significant evidence is found. In Tables 15–17 the result of the Evidence of the first row represents the comparison between the proposed Type-1 and the original methods without noise and the result of the evidence on the second row represents the comparison between the proposed type-1 and the original methods with noise.

Table 15. Results for the statistical test with Fuzzy bee colony algorithm.

| Controller | μ_1 | μ_2 | Z-Value | Evidence |
|-------------------|------------------------|-------------------|---------|----------|
| Water Tank | FBCO FLC without noise | BCO without noise | -2.734 | S |
| | FBCO FLC with noise | BCO with noise | -0.169 | S |
| Inverted Pendulum | FBCO FLC without noise | BCO without noise | -1.405 | S |
| | FBCO FLC with noise | BCO with noise | -2.600 | S |

Table 16. Results for the statistical test with Fuzzy differential evolution algorithm.

| Controller | μ_1 | μ_2 | Z-Value | Evidence |
|-------------------|-----------------------|------------------|---------|----------|
| Water Tank | FDE FLC without noise | DE without noise | -0.0655 | N.S |
| | FDE FLC with noise | DE with noise | -24.213 | S |
| Inverted Pendulum | FDE FLC without noise | DE without noise | -7.525 | S |
| | FDE FLC with noise | DE with noise | -2.335 | S |

Table 17. Results for the statistical test with fuzzy harmony search algorithm.

| Controller | μ_1 | μ_2 | Z-Value | Evidence |
|-------------------|-----------------------|------------------|---------|----------|
| Water Tank | FHS FLC without noise | HS without noise | −1.5018 | N.S |
| | FHS FLC with noise | HS with noise | −1.79 | S |
| Inverted Pendulum | FHS FLC without noise | HS without noise | −4.25 | S |
| | FHS FLC with noise | HS with noise | −2.40 | S |

The results of the statistical tests show that for most of the cases the proposal used for each of the algorithms is good, however, the important point of this paper is the comparison of the three methods, with respect to the BCO algorithm the six statistical tests have significant evidence, presenting greater problem with the inverted pendulum controller.

For the case of the DE algorithm of six statistical tests only in five has significant evidence, having problem in the water tank controller, and finally the HS algorithm of six statistical tests four of them have significant evidence, failing both the water tank and inverted pendulum controllers. Statistically we can say that the algorithm BCO is better compared to the DE and HS algorithms.

7. Discussion

Every algorithm that has been analyzed required the optimal parameters to minimize the error and achieve the stability in the two problems presented. Is necessary realized several experiments to meet these parameters. In this research, we realized a study to determine the optimal algorithm with a better error in the simulations, this will allow to have a generalization of which algorithm allows to find better errors in control problems, in this case FHS presents excellent results compared to FBCO and FDE respect to average and standard deviation, but compare to the best error FDE is better.

Once the experiments in Section 5 are shown, we analyzed the following results; for the first problems studied for the water tank, Figure 11 (part A) shows that for the original and fuzzy DE the lower errors are found in initial iterations, these results are reflected in Tables 6, 8 and 10; the best results for the FBCO was 3.79×10^{-1} , for the FDE was 4.82×10^{-2} , and for FHS was 4.86×10^{-1} , the results are similar but, a significant difference is shown with FDE. Otherwise, when noise is added in the model the DE continues to have lower errors (see Figure 13 part A), the best in FBCO was 3.80×10^{-1} , for FDE was 1.27×10^{-3} and for FHS was 2.53×10^{-2} .

Another important metric to compare is the standard deviation, and this allows to shows if the values are closer in the simulation. For example, for the second problem statement, FBCO without noise the value was 4.66×10^{-1} , for FDE was 2.75×10^{-1} and for FHS was 4.81×10^{-1} , and respect to average; FBCO was 8.21×10^{-2} , for FDE was 2.15×10^{-1} and for FHS was 7.67×10^{-1} , in this case FDE is more stable.

8. Conclusions

Differential evolution is a widely used algorithm, in literature we can find various works using DE and it has shown good performance in different applications. HS and BCO algorithms are relatively new in the field of research they have shown promising results in different types of problems. There is no work in which these three algorithms (DE, HS, and BCO) have been used to make a comparison with control problems, the idea of conducting this work was to compare the performance of the three algorithms. The main contribution of this paper is the improvement that is made to each of the algorithms using fuzzy logic, which dynamically moves the parameters of the membership functions of the control problem. Each of the statistical tests by each algorithm has shown that the use of fuzzy logic in the algorithms is better than the original algorithm.

As we can note in Figures 17–19 the performance for each of the algorithms in its three phases of experimentation, we can confirm that the performance of the algorithms using fuzzy logic improves as the complexity of the problem increases. For this study that we carried out, the differential evolution

confirms the aforementioned, and the performance of the FDE algorithm compared to the other two algorithms used FHS and FBCO was better for both control problems. Additionally, the time performance in seconds of each method for each type of problem is shown and it can be seen that in some cases the proposed method takes longer but always achieves better results.

In summary, the FDE algorithm outperformed the other two algorithms (FHS and FBCO) for both control problems. The performance in each algorithms is different, for example, in the methodology of DE algorithm allows to better evaluate the individuals in each generation, this is because it performs the mutation and crossing operations, which allow improving the individuals of the population, a feature that BCO and HS does not present. In general, we presented an idea of the performance of each method for the case studies shown in this article, and that can serve as a basis for other works in the future.

For future work, it is intended to implement to using interval and generalized type-2 fuzzy logic system to handle a higher level of uncertainty. Additionally, the proposed algorithms can be compared with the use of other applications, fuzzy systems applied to other problems and the optimization of neural networks. Similarly include a comparison with more metaheuristics with the same characteristics in order to apply the proposed methodology to various kinds of control problems.

Author Contributions: J.S. analyzed the results; O.C. and F.V. conceived of the presented idea and developed the framework of this study; C.P., P.O., and L.A.-A. carried out the experiments and wrote the article. All authors discussed the results and contributed to the final manuscript.

Acknowledgments: We thank the Division of Graduate Studies and Research of Tijuana Institute of Technology and the financial support provided by CONACYT contract grant Number: 122.

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

References

1. 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](#)]
2. Amador-Angulo, L.; Mendoza, O.; Castro, J.; Rodríguez-Díaz, A.; Melin, P.; Castillo, O. Fuzzy Sets in Dynamic Adaptation of Parameters of a Bee Colony Optimization for Controlling the Trajectory of an Autonomous Mobile Robot. *Sensors* **2016**, *16*, 1458. [[CrossRef](#)] [[PubMed](#)]
3. Vasant, P.; Alparslan-Gok, S.Z.; Weber, G.-W. (Eds.) *Handbook of Research on Emergent Applications of Optimization Algorithms*; IGI Global: Hershey, PA, USA, 2018.
4. Tien Bui, D.; Anh Tuan, T.; Hoang, N.; Quoc Thanh, N.; Nguyen, D.; Van Liem, N.; Pradhan, B. Spatial prediction of rainfall-induced landslides for the Lao Cai area (Vietnam) using a hybrid intelligent approach of least squares support vector machines inference model and artificial bee colony optimization. *Landslides* **2017**, *14*, 447–458. [[CrossRef](#)]
5. Castillo, O.; Amador-Angulo, L. A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. *Inf. Sci.* **2017**. [[CrossRef](#)]
6. Dell’Orco, M.; Marinelli, M.; Altieri, M.G. Solving the gate assignment problem through the Fuzzy Bee Colony Optimization. *Transp. Res. Part C Emerg. Technol.* **2017**, *80*, 424–438. [[CrossRef](#)]
7. Jovanović, A.; Nikolić, M.; Teodorović, D. Area-wide urban traffic control: A Bee Colony Optimization approach. *Transp. Res. Part C Emerg. Technol.* **2017**, *77*, 329–350. [[CrossRef](#)]
8. Konar, M.; Bagis, A. Performance Comparison of Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony Algorithms for Fuzzy Modelling of Nonlinear Systems. *Elektron. Elektrotech.* **2016**, *22*, 8–13. [[CrossRef](#)]
9. Yadav, N.; Ngo, T.T.; Yadav, A.; Kim, J.H. Numerical Solution of Boundary Value Problems Using Artificial Neural Networks and Harmony Search. In *Harmony Search Algorithm*; Del Ser, J., Ed.; Springer Singapore: Singapore, 2017; Volume 514, pp. 112–118.
10. Mazare, M.; Taghizadeh, M.; Kazemi, M.G. Optimal hybrid scheme of dynamic neural network and PID controller based on harmony search algorithm to control a PWM-driven pneumatic actuator position. *J. Vib. Control* **2017**. [[CrossRef](#)]

11. Kattan, A.; Abdullah, R.; Salam, R.A. Harmony Search Based Supervised Training of Artificial Neural Networks. In Proceedings of the 2010 International Conference on Intelligent Systems, Modelling and Simulation, Liverpool, UK, 27–29 January 2010; pp. 105–110.
12. Peraza, C.; Valdez, F.; Castillo, O. Study on the Use of Type-1 and Interval Type-2 Fuzzy Systems Applied to Benchmark Functions Using the Fuzzy Harmony Search Algorithm. In *Fuzzy Logic in Intelligent System Design*; Melin, P., Castillo, O., Kacprzyk, J., Reformat, M., Melek, W., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 648, pp. 94–103.
13. Peraza, C.; Valdez, F.; Garcia, M.; Melin, P.; Castillo, O. A New Fuzzy Harmony Search Algorithm Using Fuzzy Logic for Dynamic Parameter Adaptation. *Algorithms* **2016**, *9*, 69. [[CrossRef](#)]
14. Peraza, C.; Valdez, F.; Castillo, O. An Adaptive Fuzzy Control Based on Harmony Search and Its Application to Optimization. In *Nature-Inspired Design of Hybrid Intelligent Systems*; Melin, P., Castillo, O., Kacprzyk, J., Eds.; Springer International Publishing: Cham, Switzerland, 2017; Volume 667, pp. 269–283.
15. Peraza, C.; Valdez, F.; Melin, P. Optimization of Intelligent Controllers Using a Type-1 and Interval Type-2 Fuzzy Harmony Search Algorithm. *Algorithms* **2017**, *10*, 82. [[CrossRef](#)]
16. Gao, K.Z.; Suganthan, P.N.; Pan, Q.K.; Chua, T.J.; Cai, T.X.; Chong, C.S. Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives. *J. Intell. Manuf.* **2016**, *27*, 363–374. [[CrossRef](#)]
17. Prayogo, D.; Cheng, M.Y.; Prayogo, H. A Novel Implementation of Nature-inspired Optimization for Civil Engineering: A Comparative Study of Symbiotic Organisms Search. *Civ. Eng. Dimens.* **2017**, *19*.
18. Ouyang, H.; Ma, G.; Liu, G.; Li, Z.; Zhong, X. Hybrid teaching-learning based optimization with harmony search for engineering optimization problems. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 2714–2717.
19. Goudos, S.K.; Deruyck, M.; Plets, D.; Martens, L.; Joseph, W. Optimization of Power Consumption in 4G LTE Networks Using a Novel Barebones Self-adaptive Differential Evolution Algorithm. *Telecommun. Syst.* **2017**, *66*, 109–120. [[CrossRef](#)]
20. Zheng, L.; Peng, D.; Sun, Y.; Gao, S. Research of Model Identification for Control System Based on Improved Differential Evolution Algorithm. In *Advanced Computational Methods in Life System Modeling and Simulation*; Fei, M., Ma, S., Li, X., Sun, X., Jia, L., Su, Z., Eds.; Springer Singapore: Singapore, 2017; Volume 761, pp. 282–293.
21. Ochoa, P.; Castillo, O.; Soria, J. Differential Evolution Using Fuzzy Logic and a Comparative Study with Other Metaheuristics. In *Nature-Inspired Design of Hybrid Intelligent Systems*; Springer: Cham, Switzerland, 2017; pp. 257–268.
22. Ochoa, P.; Castillo, O.; Soria, J. Type-2 fuzzy logic dynamic parameter adaptation in a new Fuzzy Differential Evolution method. In Proceedings of the 2016 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS), El Paso, TX, USA, 31 October–4 November 2016; pp. 1–6.
23. Salehpour, M.; Jamali, A.; Bagheri, A.; Nariman-zadeh, N. A new adaptive differential evolution optimization algorithm based on fuzzy inference system. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 587–597. [[CrossRef](#)]
24. Beirami, H.; Shabestari, A.Z.; Zerafat, M.M. Optimal PID plus fuzzy controller design for a PEM fuel cell air feed system using the self-adaptive differential evolution algorithm. *Int. J. Hydrogen Energy* **2015**, *40*, 9422–9434. [[CrossRef](#)]
25. Zhang, X.; Zhang, X. Shift based adaptive differential evolution for PID controller designs using swarm intelligence algorithm. *Clust. Comput.* **2017**, *20*, 291–299. [[CrossRef](#)]
26. Maurya, R.; Bhandari, M. Fractional Order PID Controller with an Improved Differential Evolution Algorithm. In Proceedings of the 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), Ghaziabad, India, 22–23 September 2016; pp. 550–554.
27. 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](#)]
28. 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](#)]
29. Pamucar, D.; Atanaskovic, P.; Milicic, M. Modeling of fuzzy logic system for investment management in the railway infrastructure/Modeliranje neizrazitog (fuzzy) logickog sustava za upravljanje investicijama na zeljeznickoj infrastrukturi. *Teh. Vjesn. Tech. Gaz.* **2015**, *22*, 1185–1193.

30. Pamučar, D.; Vasin, L.; Atanasković, P.; Miličić, M. Planning the City Logistics Terminal Location by Applying the Green-Median Model and Type-2 Neurofuzzy Network. *Comput. Intell. Neurosci.* **2016**. [[CrossRef](#)]
31. Pamučar, D.S.; Božanić, D.; Komazec, N. Risk assessment of natural disasters using fuzzy logic system of type 2. *Manag. J. Sustain. Bus. Manag. Solut. Emerg. Econ.* **2017**, *21*, 23–34.
32. Sremac, S.; Tanackov, I.; Kopic, M.; Radović, D. ANFIS model for determining the economic order quantity. *Decis. Mak. Appl. Manag. Eng.* **2018**, *1*, 1–12. [[CrossRef](#)]
33. Mamdani, E.H. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. Inst. Electr. Eng.* **1974**, *121*, 1585. [[CrossRef](#)]
34. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning—I. *Inf. Sci.* **1975**, *8*, 199–249. [[CrossRef](#)]
35. Kothamasu, R.; Huang, S.H. Adaptive Mamdani fuzzy model for condition-based maintenance. *Fuzzy Sets Syst.* **2007**, *158*, 2715–2733. [[CrossRef](#)]
36. Akkurt, S.; Tayfur, G.; Can, S. Fuzzy logic model for the prediction of cement compressive strength. *Cem. Concr. Res.* **2004**, *34*, 1429–1433. [[CrossRef](#)]
37. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *SMC-15*, 116–132. [[CrossRef](#)]
38. Cao, Y.-Y.; Frank, P.M. Stability analysis and synthesis of nonlinear time-delay systems via linear Takagi–Sugeno fuzzy models. *Fuzzy Sets Syst.* **2001**, *124*, 213–229. [[CrossRef](#)]
39. Cervantes, L.; Castillo, O.; Melin, P.; Valdez, F. Comparative Study of Type-1 and Type-2 Fuzzy Systems for the Three-Tank Water Control Problem. In *Advances in Computational Intelligence*; Batyrshin, I., Mendoza, M.G., Eds.; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2013; Volume 7630, pp. 362–373.
40. Castillo, O.; Amador-Angulo, L.; Castro, J.R.; Garcia-Valdez, M. A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. *Inf. Sci.* **2016**, *354*, 257–274. [[CrossRef](#)]
41. El-Hawwary, M.I.; Elshafei, A.L.; Emara, H.M.; Fattah, H.A.A. Adaptive Fuzzy Control of the Inverted Pendulum Problem. *IEEE Trans. Control Syst. Technol.* **2006**, *14*, 1135–1144. [[CrossRef](#)]
42. Ontiveros, E.; Melin, P.; Castillo, O. High order α -planes integration: A new approach to computational cost reduction of General Type-2 Fuzzy Systems. *Eng. Appl. Artif. Intell.* **2018**, *74*, 186–197. [[CrossRef](#)]
43. Lu, P.; Zhou, J.; Zhang, H.; Zhang, R.; Wang, C. Chaotic differential bee colony optimization algorithm for dynamic economic dispatch problem with valve-point effects. *Int. J. Electr. Power Energy Syst.* **2014**, *62*, 130–143. [[CrossRef](#)]
44. Sharma, V.; Srinivasan, K.; Kumar, R.; Chao, H.-C.; Hua, K.-L. Efficient cooperative relaying in flying ad hoc networks using fuzzy-bee colony optimization. *J. Supercomput.* **2017**, *73*, 3229–3259. [[CrossRef](#)]
45. Moayedikia, A.; Jensen, R.; Wiil, U.K.; Forsati, R. Weighted bee colony algorithm for discrete optimization problems with application to feature selection. *Eng. Appl. Artif. Intell.* **2015**, *44*, 153–167. [[CrossRef](#)]
46. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
47. Manjarres, D.; Landa-Torres, I.; Gil-Lopez, S.; Del Ser, J.; Bilbao, M.N.; Salcedo-Sanz, S.; Geem, Z.W. A survey on applications of the harmony search algorithm. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1818–1831. [[CrossRef](#)]
48. Geem, Z.W. State-of-the-Art in the Structure of Harmony Search Algorithm. In *Recent Advances in Harmony Search Algorithm*; Geem, Z.W., Ed.; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2010; Volume 270, pp. 1–10.

