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

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 \([0,1]\):

\[
A = \{(x, u_A(x)) \mid x \in U\}
\]  

(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.](image)

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.

\[
R^l : \text{IF } x_1 \text{ is } F_1^l \text{ and } \ldots \text{ and } x_p \text{ is } F_p^l \text{ THEN } y \text{ is } G^l, \text{ where } l = 1, \ldots, M
\]  

(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.

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

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

(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 2. Water tank system.](image_url)

![Figure 3. Water tank system.](image_url)
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, \( \theta \) 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.](image)

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.

![Figure 6.](image)

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

\[
\text{Iteration} = \frac{\text{Current iteration}}{\text{Maximum of iterations}} \quad (4)
\]

\[
\text{Diversity } (S(t)) = \frac{1}{n_S} \sum_{i=1}^{n_S} \left( \sum_{j=1}^{n_S} (x_{ij}(t) - \bar{x}_j(t))^2 \right)^{1/2} \quad (5)
\]

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:

\[ P_{ij,n} = \frac{[p_{ij,n}]^a \cdot \left[ \frac{1}{d_{ij}} \right]^\beta}{\sum_{j \in A_i} [p_{ij,n}]^a \cdot \left[ \frac{1}{d_{ij}} \right]^\beta} \]  \hspace{1cm} (6)

\[ D_i = K \cdot \frac{P_{fi}}{P_{f_{colony}}} \]  \hspace{1cm} (7)

\[ P_{fi} = \frac{1}{L_i}, L_i = \text{Tour Length} \]  \hspace{1cm} (8)

\[ P_{f_{colony}} = \frac{1}{N_{Bee}} \sum_{i=1}^{N_{Bee}} P_{fi} \]  \hspace{1cm} (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.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Operator</th>
<th>Diversity</th>
<th>Beta</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>and</td>
<td>Low</td>
<td>then</td>
<td>High</td>
</tr>
<tr>
<td>Low</td>
<td>and</td>
<td>Medium</td>
<td>then</td>
<td>Medium High</td>
</tr>
<tr>
<td>Low</td>
<td>and</td>
<td>High</td>
<td>then</td>
<td>Medium High</td>
</tr>
<tr>
<td>Medium</td>
<td>and</td>
<td>Low</td>
<td>then</td>
<td>Medium High</td>
</tr>
<tr>
<td>Medium</td>
<td>and</td>
<td>Medium</td>
<td>then</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>and</td>
<td>Low</td>
<td>then</td>
<td>Medium Low</td>
</tr>
<tr>
<td>High</td>
<td>and</td>
<td>Low</td>
<td>then</td>
<td>Medium Low</td>
</tr>
<tr>
<td>High</td>
<td>and</td>
<td>Medium</td>
<td>then</td>
<td>Medium Low</td>
</tr>
<tr>
<td>High</td>
<td>and</td>
<td>High</td>
<td>then</td>
<td>Low</td>
</tr>
</tbody>
</table>

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 new solutions
                    Make a decision whether the best is royal
                    If the bee is not royal then
                        Choice one of the royal bees followed by the i-th 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].
4.2. 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:**

\[ P_{x,g} = (x_{i,g}), \quad i = 0, 1, \ldots, N_p - 1, \quad g = 0, 1, \ldots, g_{\text{max}} \]

\[ x_{i,g} = (x_{j,i,g}), \quad j = 0, 1, \ldots, D - 1 \]  \hspace{1cm} (10)

\[ P_{v,g} = (v_{i,g}), \quad i = 0, 1, \ldots, N_p - 1, \quad g = 0, 1, \ldots, g_{\text{max}} \]

\[ v_{i,g} = (v_{j,i,g}), \quad j = 0, 1, \ldots, D - 1 \]  \hspace{1cm} (11)

\[ P_{u,g} = (u_{i,g}), \quad i = 0, 1, \ldots, N_p - 1, \quad g = 0, 1, \ldots, g_{\text{max}} \]

\[ u_{i,g} = (u_{j,i,g}), \quad j = 0, 1, \ldots, D - 1 \]  \hspace{1cm} (12)

**Initialization:**

\[ x_{j,0} = \text{rand}_j(0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L} \]  \hspace{1cm} (13)

**Mutation:**

\[ v_{i,g} = \text{rand}_g \cdot F \cdot (x_{1,g} - x_{2,g}) \]  \hspace{1cm} (14)

**Crossover:**

\[ u_{i,g} = u_{j,i,g} \begin{cases} v_{i,g} & \text{if } (\text{rand}_j(0,1) \leq C_r \text{ or } j = j_{\text{rand}}) \\ x_{j,i,g} & \text{otherwise} \end{cases} \]  \hspace{1cm} (15)

**Selection:**

\[ x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases} \]  \hspace{1cm} (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.](image)

**Table 2. Fuzzy rules for the FDE.**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generations</td>
<td>Operator</td>
</tr>
<tr>
<td>Low and Low then High</td>
<td>Low</td>
</tr>
<tr>
<td>Low and Medium then High</td>
<td>Low</td>
</tr>
<tr>
<td>Low and High then Medium high</td>
<td>Medium Low</td>
</tr>
<tr>
<td>Medium and Low then Medium High</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium and Medium then Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium and High then Medium Low</td>
<td>Medium Low</td>
</tr>
<tr>
<td>High and Low then Medium Low</td>
<td>Medium Low</td>
</tr>
<tr>
<td>High and Medium then Medium Low</td>
<td>Medium Low</td>
</tr>
</tbody>
</table>

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).
Generate the initial population of individuals

Do
    For each individual j in the population
        Choose three numbers $x_1$, $x_2$ and $x_3$ that is, $1 \leq x_0$, $x_1$, $x_2 \leq N$ with $x_0 \neq x_1 \neq x_2 \neq j$
    Generate a random integer $i_{rand} \in (1, N)$
    For each parameter $i$
        $v_{i,g} = x_{r_1,g} + F \cdot (x_{r_2,g} - x_{r_3,g})$
        $u_{i,g} = u_{i,g}(v_{i,g} \text{ if } (rand_i(0,1) \leq C_r \text{ or } j = j_{rand})$
        $x_{i,g} \text{ otherwise}$
    End For
    Replace $x_{j,i,g}$ with the child $u_{i,g}$ if $u_{i,g}$ is better
End For
Until the termination condition is achieved

Figure 12. Differential evolution algorithm.

Generate the initial population of individuals

Do
    For each individual j in the population
        Choose three numbers $x_1$, $x_2$ and $x_3$ that is, $1 \leq x_0$, $x_1$, $x_2 \leq N$ with $x_0 \neq x_1 \neq x_2 \neq j$
    Generate a random integer $i_{rand} \in (1, N)$
    For each parameter $i$
        Calculate generation and diversity using Equations (4) and (5)
        Use a fuzzy system to calculate the new Mutation and Crossover parameters
        $v_{i,g} = x_{r_1,g} + F \cdot (x_{r_2,g} - x_{r_3,g})$
        $u_{i,g} = u_{i,g}(v_{i,g} \text{ if } (rand_i(0,1) \leq C_r \text{ or } j = j_{rand})$
        $x_{i,g} \text{ otherwise}$
    End For
    Replace $x_{j,i,g}$ with the child $u_{i,g}$ if $u_{i,g}$ is better
End For
Until the termination condition is achieved

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]:

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

$$P_{Arate} = P_{lower \ limit} + P_{range} \cdot rand$$  \hfill (18)

$$P_{range} = P_{upper \ limit} - P_{lower \ limit}$$  \hfill (19)

$$P_{random} = 1 - HMR$$  \hfill (20)

$$P_{pitch} = HMR \cdot P_{Arate}$$  \hfill (21)
where \( x_{\text{old}} \) is the current solution, \( x_{\text{new}} \) is the new solution, and \( \text{rand} \) 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):

\[
\begin{align*}
    x_{\text{new}} &= x_{\text{old}} + b \cdot (2 \cdot \text{rand} - 1) \\
    P_{\text{HMR}} &= P_{\text{HMR}}^\text{old} + P_{\text{HMR}}^\text{new} \cdot r \cdot a \cdot n \cdot d \\
    P_{\text{PARate}} &= 1 - H_{\text{MAR}} \\
    P_{\text{PARate}} &= H_{\text{MAR}} \cdot P_{\text{ARate}}^\text{old} + P_{\text{ARate}}^\text{new}
\end{align*}
\]

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.
Figure 15. Harmony search algorithm.

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>BCO</th>
<th>DE</th>
<th>HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>NP(Population)</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>FoodNumber</td>
<td>NP/2</td>
<td>F (mutation)</td>
<td>Harmonies</td>
</tr>
<tr>
<td>Alpha</td>
<td>0.5</td>
<td>CR (crossover)</td>
<td>0.3</td>
</tr>
<tr>
<td>Beta</td>
<td>2.5</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5. Parameters used with the proposed methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>FBCO</th>
<th>FDE</th>
<th>FHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>NP(Population)</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>FoodNumber</td>
<td>NP/2</td>
<td>F (mutation)</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Alpha</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Beta</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>

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):

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

(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.

<table>
<thead>
<tr>
<th>Water Tank Controller</th>
<th>BCO</th>
<th>FBCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>FLC without Noise</td>
<td>FLC with Noise</td>
</tr>
<tr>
<td>Best</td>
<td>$4.50 \times 10^{-1}$</td>
<td>$3.64 \times 10^{-1}$</td>
</tr>
<tr>
<td>Worst</td>
<td>$5.91 \times 10^{-1}$</td>
<td>$5.06 \times 10^{-1}$</td>
</tr>
<tr>
<td>Average</td>
<td>$5.21 \times 10^{-1}$</td>
<td>$4.50 \times 10^{-1}$</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>$3.47 \times 10^{-1}$</td>
<td>$2.98 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>RMSE</th>
<th>BCO FLC without Noise</th>
<th>BCO FLC with Noise</th>
<th>FBCO FLC without Noise</th>
<th>FBCO FLC with Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>$4.37 \times 10^{-1}$</td>
<td>$3.78 \times 10^{-1}$</td>
<td>$3.79 \times 10^{-1}$</td>
<td>$3.80 \times 10^{-1}$</td>
</tr>
<tr>
<td>Worst</td>
<td>$5.61 \times 10^{-1}$</td>
<td>$8.64 \times 10^{-1}$</td>
<td>$8.77 \times 10^{-1}$</td>
<td>$9.47 \times 10^{-1}$</td>
</tr>
<tr>
<td>Average</td>
<td>$5.07 \times 10^{-2}$</td>
<td>$6.02 \times 10^{-1}$</td>
<td>$8.21 \times 10^{-2}$</td>
<td>$4.82 \times 10^{-1}$</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>$3.23 \times 10^{-1}$</td>
<td>$3.45 \times 10^{-1}$</td>
<td>$4.66 \times 10^{-1}$</td>
<td>$9.19 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>RMSE</th>
<th>DE FLC Without Noise</th>
<th>DE FLC With Noise</th>
<th>FDE FLC Without Noise</th>
<th>FDE FLC with Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>$1.65 \times 10^{-2}$</td>
<td>$1.08 \times 10^{-1}$</td>
<td>$4.82 \times 10^{-2}$</td>
<td>$1.27 \times 10^{-3}$</td>
</tr>
<tr>
<td>Worst</td>
<td>$2.36 \times 10^{-1}$</td>
<td>$4.87 \times 10^{-1}$</td>
<td>$1.82 \times 10^{-1}$</td>
<td>$2.32 \times 10^{-1}$</td>
</tr>
<tr>
<td>Average</td>
<td>$9.37 \times 10^{-2}$</td>
<td>$4.48 \times 10^{-1}$</td>
<td>$9.44 \times 10^{-2}$</td>
<td>$3.67 \times 10^{-2}$</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>$4.81 \times 10^{-2}$</td>
<td>$7.43 \times 10^{-2}$</td>
<td>$3.34 \times 10^{-2}$</td>
<td>$5.60 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>RMSE</th>
<th>DE FLC without Noise</th>
<th>DE FLC with Noise</th>
<th>FDE FLC without Noise</th>
<th>FDE FLC with Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>$8.45 \times 10^{-2}$</td>
<td>$2.96 \times 10^{-1}$</td>
<td>$1.83 \times 10^{-2}$</td>
<td>$1.40 \times 10^{-2}$</td>
</tr>
<tr>
<td>Worst</td>
<td>$1.46 \times 10^{0}$</td>
<td>$2.99 \times 10^{-1}$</td>
<td>$1.40 \times 10^{0}$</td>
<td>$7.92 \times 10^{-1}$</td>
</tr>
<tr>
<td>Average</td>
<td>$8.99 \times 10^{-1}$</td>
<td>$2.97 \times 10^{-1}$</td>
<td>$2.15 \times 10^{-1}$</td>
<td>$3.78 \times 10^{-1}$</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>$4.15 \times 10^{-1}$</td>
<td>$6.28 \times 10^{-4}$</td>
<td>$2.75 \times 10^{-1}$</td>
<td>$1.90 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>RMSE</th>
<th>HS FLC without Noise</th>
<th>HS FLC with Noise</th>
<th>FHS FLC without Noise</th>
<th>FHS FLC with Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>$4.98 \times 10^{-1}$</td>
<td>$1.01 \times 10^{-2}$</td>
<td>$4.86 \times 10^{-1}$</td>
<td>$2.53 \times 10^{-2}$</td>
</tr>
<tr>
<td>Worst</td>
<td>$6.24 \times 10^{-1}$</td>
<td>$6.99 \times 10^{-1}$</td>
<td>$6.25 \times 10^{-1}$</td>
<td>$2.01 \times 10^{-1}$</td>
</tr>
<tr>
<td>Average</td>
<td>$5.66 \times 10^{-1}$</td>
<td>$1.40 \times 10^{-1}$</td>
<td>$5.54 \times 10^{-1}$</td>
<td>$8.36 \times 10^{-2}$</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>$3.26 \times 10^{-2}$</td>
<td>$1.68 \times 10^{-1}$</td>
<td>$2.92 \times 10^{-2}$</td>
<td>$3.52 \times 10^{-2}$</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>RMSE</th>
<th>HS FLC without Noise</th>
<th>HS FLC with Noise</th>
<th>FHS FLC without Noise</th>
<th>FHS FLC with Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>$3.15 \times 10^{-1}$</td>
<td>$3.97 \times 10^{-1}$</td>
<td>$2.99 \times 10^{-1}$</td>
<td>$2.97 \times 10^{-1}$</td>
</tr>
<tr>
<td>Worst</td>
<td>$4.88 \times 10^{0}$</td>
<td>$1.28 \times 10^{0}$</td>
<td>$2.01 \times 10^{0}$</td>
<td>$1.76 \times 10^{-1}$</td>
</tr>
<tr>
<td>Average</td>
<td>$1.88 \times 10^{0}$</td>
<td>$1.02 \times 10^{0}$</td>
<td>$7.67 \times 10^{-1}$</td>
<td>$7.54 \times 10^{-1}$</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>$1.35 \times 10^{0}$</td>
<td>$4.07 \times 10^{-1}$</td>
<td>$4.81 \times 10^{-1}$</td>
<td>$4.49 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

**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 18](image)

**Figure 18.** 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Original FLC without Noise</th>
<th>Original FLC with Noise</th>
<th>Proposed FLC without Noise</th>
<th>Proposed FLC with Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCO</td>
<td>381.603</td>
<td>515.283</td>
<td>408.816</td>
<td>495.371</td>
</tr>
<tr>
<td>DE</td>
<td>205.357</td>
<td>237.064</td>
<td>228.372</td>
<td>268.867</td>
</tr>
<tr>
<td>HS</td>
<td>890.95</td>
<td>65.63</td>
<td>885.70</td>
<td>77.16</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Original without Noise</th>
<th>Original with Noise</th>
<th>Proposed without Noise</th>
<th>Proposed with Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCO</td>
<td>323.84</td>
<td>298.92</td>
<td>492.03</td>
<td>682.54</td>
</tr>
<tr>
<td>DE</td>
<td>310.66</td>
<td>380.76</td>
<td>474.44</td>
<td>682.14</td>
</tr>
<tr>
<td>HS</td>
<td>316.08</td>
<td>2326.63</td>
<td>508.74</td>
<td>1173.25</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level of Confidence</td>
<td>95%</td>
</tr>
<tr>
<td>Alpha</td>
<td>0.05%</td>
</tr>
<tr>
<td>( H_0 )</td>
<td>( \mu_1 \geq \mu_2 )</td>
</tr>
<tr>
<td>( H_a )</td>
<td>( \mu_1 &lt; \mu_2 )</td>
</tr>
<tr>
<td>Critical Value</td>
<td>−1.645</td>
</tr>
</tbody>
</table>

The hypotheses used for the statistical test indicate the following:

\( H_0 \): 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:

\[
Z = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sigma_{\bar{X}_1 - \bar{X}_2}}
\]  

(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.

<table>
<thead>
<tr>
<th>Controller</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>Z-Value</th>
<th>Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water Tank</td>
<td>FBCO FLC without noise</td>
<td>BCO without noise</td>
<td>−2.734</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>FBCO FLC with noise</td>
<td>BCO with noise</td>
<td>−0.169</td>
<td>S</td>
</tr>
<tr>
<td>Inverted Pendulum</td>
<td>FBCO FLC without noise</td>
<td>BCO without noise</td>
<td>−1.405</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>FBCO FLC with noise</td>
<td>BCO with noise</td>
<td>−2.600</td>
<td>S</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Controller</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>Z-Value</th>
<th>Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water Tank</td>
<td>FDE FLC without noise</td>
<td>DE without noise</td>
<td>−0.0655</td>
<td>N.S</td>
</tr>
<tr>
<td></td>
<td>FDE FLC with noise</td>
<td>DE with noise</td>
<td>−24.213</td>
<td>S</td>
</tr>
<tr>
<td>Inverted Pendulum</td>
<td>FDE FLC without noise</td>
<td>DE without noise</td>
<td>−7.525</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>FDE FLC with noise</td>
<td>DE with noise</td>
<td>−2.335</td>
<td>S</td>
</tr>
</tbody>
</table>
Table 17. Results for the statistical test with fuzzy harmony search algorithm.

<table>
<thead>
<tr>
<th>Controller</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>Z-Value</th>
<th>Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water Tank</td>
<td>FHS FLC without noise</td>
<td>HS without noise</td>
<td>−1.5018</td>
<td>N.S</td>
</tr>
<tr>
<td></td>
<td>FHS FLC with noise</td>
<td>HS with noise</td>
<td>−1.79</td>
<td>S</td>
</tr>
<tr>
<td>Inverted Pendulum</td>
<td>FHS FLC without noise</td>
<td>HS without noise</td>
<td>−4.25</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>FHS FLC with noise</td>
<td>HS with noise</td>
<td>−2.40</td>
<td>S</td>
</tr>
</tbody>
</table>

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 $3.79 \times 10^{-1}$, for the FDE was $4.82 \times 10^{-2}$, and for FHS was $4.86 \times 10^{-1}$, 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 $3.80 \times 10^{-1}$, for FDE was $1.27 \times 10^{-3}$ and for FHS was $2.53 \times 10^{-2}$.

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 $4.66 \times 10^{-1}$, for FDE was $2.75 \times 10^{-1}$ and for FHS was $4.81 \times 10^{-1}$, and respect to average; FBCO was $8.21 \times 10^{-2}$, for FDE was $2.15 \times 10^{-1}$ and for FHS was $7.67 \times 10^{-1}$, 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**


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]


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]


34. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning—I. Inf. Sci. 1975, 8, 199–249. [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]

