Hindawi
Advances in Fuzzy Systems
Volume 2018, Article ID 1274969, 19 pages
https://doi.org/10.1155/2018/1274969

Research Article

Fuzzy Dynamic Parameter Adaptation in ACO and PSO for Designing Fuzzy Controllers: The Cases of Water Level and Temperature Control

Fevrier Valdez,[1] **Juan Carlos Vazquez**,[1] **and Fernando Gaxiola**[2]

[1]*Tijuana Institute of Technology, Tijuana, BC, Mexico*
[2]*Autonomous University of Chihuahua, Chihuahua, CHIH, Mexico*

Correspondence should be addressed to Fevrier Valdez; fevrier@tectijuana.mx

Received 19 January 2018; Revised 23 March 2018; Accepted 22 April 2018; Published 2 July 2018

Academic Editor: Roberto Sepúlveda

A novel approach applied to Particle Swarm Optimization (PSO) and Ant Colony Optimization is presented. The main contribution of this work is the use of fuzzy systems to dynamically update the parameters for the ACO and PSO algorithms. In the case of ACO, two fuzzy systems are designed for the Ant Colony System (ACS) algorithm variant. The first system adjusts the value for the pheromone evaporation parameter from the global pheromone trail update equation and the second system adjusts the values for the pheromone evaporation parameter from the local pheromone trail update equation. In the case of PSO, a fuzzy system is designed to find the values for the inertia weight parameter from the velocity equation. Fuzzy logic controllers (FLCs) are optimized with ACO and PSO, respectively, to prove the performance of the proposed approach. The particular benchmark problems considered to test the proposed methods are the water level control in a tank and temperature control in a shower. Therefore, PSO and ACO algorithms are applied in the optimization of the parameters of the FLCs. The achievement of the proposed fuzzy ACO and PSO algorithms is compared with the original results of each benchmark control problem.

1. Introduction

In automating the design of fuzzy controllers, the use of metaheuristic algorithms has been amply proposed. The proposed approach is based on the swarm intelligence models [1] that perform research of the collective behavior in decentralized schemes; two swarm intelligence models are utilized in this paper: the Ant Colony Optimization (ACO) [2] and Particle Swarm Optimization (PSO) [3, 4] algorithms. The development of the swarm intelligence models is founded on imitating the social behavior of living beings, particularly of insects or animals, in performing the search for the optimal solution in a solution space for a problem.

ACO is a metaheuristic algorithm based on the collaboration of artificial ants in a colony to search for an optimal solution for complex problems. In ACO algorithms, the collaboration between the artificial ants is an important part, and this consists in assigning the computational resources to artificial ants (artificial agents) by indirect communication mediated by the environment. ACO algorithms are utilized to solve many paradigmatic problems, like the Travelling Salesman Problem (TSP) [5, 6]. Different ACO variants have been used for fuzzy system controller problems models [7, 8]. Those algorithms have focused on fuzzy logic parameter tuning to find the values to each membership function, the membership functions type, the number of membership functions, and so forth [9, 10].

The inspiration of PSO is based on the social behavior of flock of birds or fishes. The procedure of PSO to encounter the optimal solution is based on the task of this animal community. In a PSO method, in the search space, a swarm of particles (individuals) is scattered. Each particle can be an aspirant solution for the problem at optimization. The PSO has been implemented, obtaining optimal results, to different optimization problems, such as benchmark functions [11, 12], finding optimal results for permutation problems [13, 14], and optimizing the training of neural networks [15, 16] and we also found optimization of fuzzy inference systems [17] and

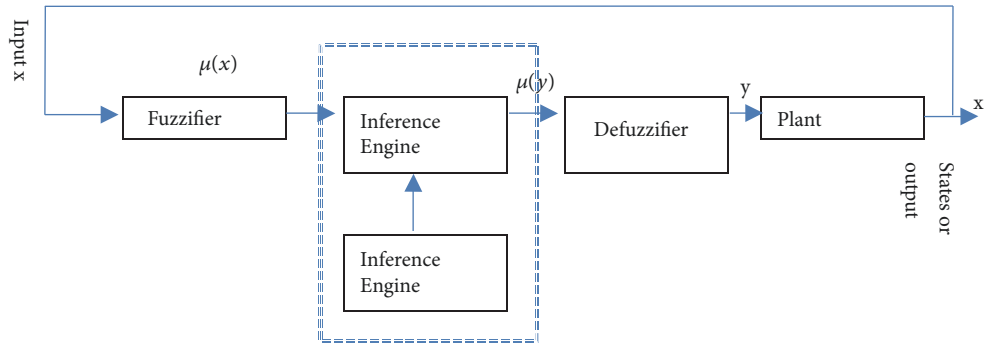

Figure 1: Scheme of fuzzy logic control (FLC).

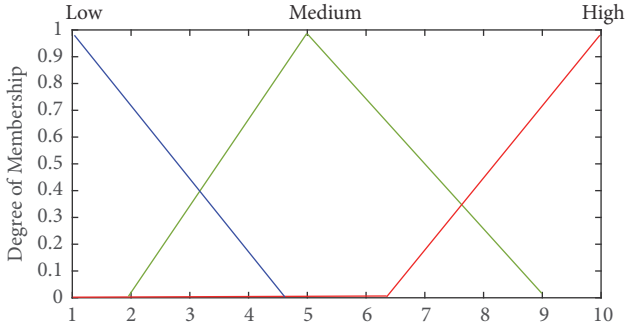

Figure 2: Membership functions for the inputs and outputs of the fuzzy inference system.

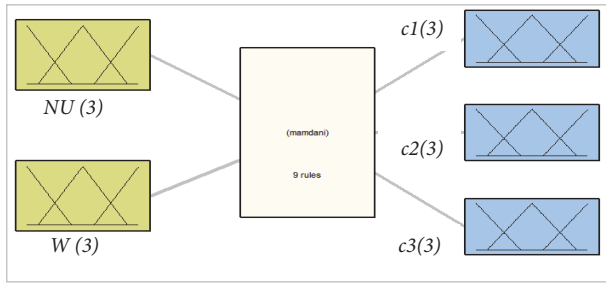

Figure 3: Fuzzy inference system to adjust the *chw*, c_1, and c_2 parameters.

Table 1: Fuzzy rules for *chw*, c_1, and c_2.

Rule number	Input		Output		
	NU	*w*	*chw*	c_1	c_2
1	1	1	3	1	1
2	1	2	1	3	3
3	1	3	1	3	3
4	2	1	3	1	1
5	2	2	2	1	3
6	2	3	1	3	3
7	3	1	3	1	1
8	3	2	1	3	3
9	3	3	1	3	3

Table 2: Fuzz rules ρ.

Rule number	Input		Output
	$\Delta\tau^{bs}$	τ	ρ
1	1	1	3
2	1	2	2
3	1	3	1
4	2	1	3
5	2	2	2
6	2	3	1
7	3	1	3
8	3	2	2
9	3	3	1

some optimization problems for fuzzy control systems [18], even with other hybrid optimization algorithms [19].

Unlike other algorithms that have been applied for control problems, we have proposed the design of fuzzy systems to adjust some parameters for each of the optimization algorithms. The algorithms, both ACO and PSO, are applied to optimize control systems [20, 21]. For this paper, we have considered two benchmark control problems for designing their fuzzy controllers with the proposed approach [22].

The novelty in the contribution of this work is the development of fuzzy systems to dynamically adjust parameters of the ACO and PSO algorithms [23, 24]. In the ACO algorithm, we have proposed the use of two fuzzy inference systems to dynamically adjust in each iteration the pheromone evaporation parameter in the equations used to calculate the update of the local and global pheromone. For the PSO algorithm, we have designed a fuzzy inference system to adjust dynamically in each iteration the parameter of the inertia weight in the equation used to obtain the velocity of the particle. The approaches were applied to the optimization of water level control and temperature control in a shower for testing.

In the PSO parameters control problems in this work, the membership functions will help in automating the fuzzy control [25, 26]. The rules were framed through numerous simulations, which are carried out to determine the best

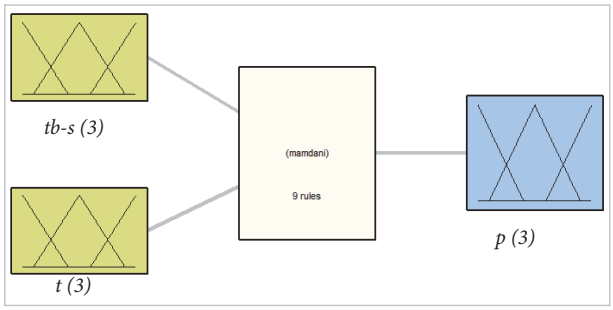

Figure 4: Fuzzy system to adjust the pheromone evaporation ρ.

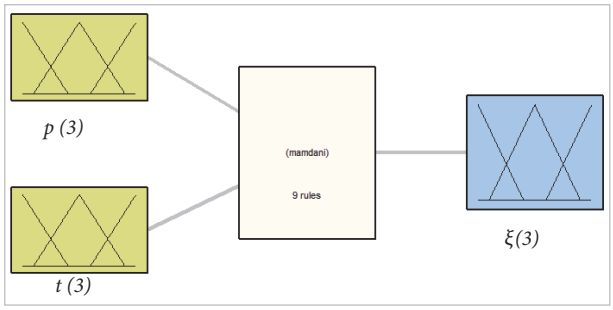

Figure 5: Fuzzy system to adjust the pheromone evaporation ξ.

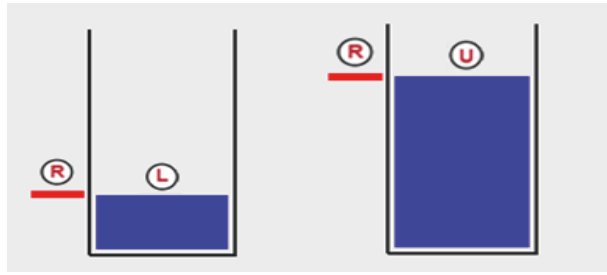

Figure 6: Water tank representation.

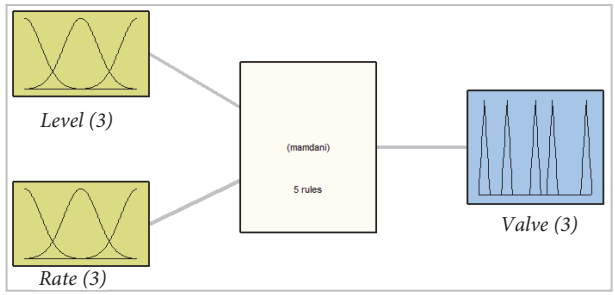

Figure 7: Original representation of the fuzzy system for water tank.

Table 3: Fuzz rules ξ.

Rule number	Input		Output
	p	τ	ξ
1	1	1	3
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	2
6	2	3	2
7	3	1	1
8	3	2	2
9	3	3	3

possible set of rules aimed at pushing the stability limits of the system to its maximum. The membership functions can be estimated by studying the behavior of different conditions and for different contingency cases [27, 28].

The organization of this paper is as follows: Section 2 describes some basic concepts of fuzzy logic, Section 3 provides a brief description of the fuzzy logic controller (FLC), and Section 3 explains the standard PSO and ACO algorithms with the basic structure for water level control in a shower in a tank and the structure for temperature control in a shower and the experimental results made in this research. The conclusions are summarized in Section 4.

2. Background

2.1. Fuzzy Logic Control (FLC). The fuzzy logic control is utilized in ill-defined complex process that can be operated by a trained human without knowing the dynamics of the system.

In a FLC, the basic idea consists in utilizing the knowledge of an expert operator for the construction of the FLC that performs the control for a system; the input-output variables for the system are represented by fuzzy rules (IF-THEN) at difference to a complicated mathematical model. In the design of the FLC, the use of fuzzy rules with linguistic variables and the fuzzy reasoning allows incorporating the experience of the human expert. A traditional scheme for a FLC is shown in Figure 1.

2.2. Particle Swarm Optimization. PSO works with particles for representing a potential solution to the optimization problem [3, 4]; the process consists of flying around a search space, and the position of each particle is updated using the personal experience and the social particles experience [29, 30]. The position of each particle is updated by the following equation:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1)$$

where $x_i(t+1)$ describes the position of the particle i in the time $(t+1)$, $x_i(t)$ describes the position of the particle i in the time (t), and $v_i(t+1)$ denotes the velocity of the particle i in the time $(t+1)$.

In this work, we utilized the gbest PSO variant, in which the experience neighborhood for each particle is obtained

from the entire swarm. For each particle i, the velocity is calculated with the equation as follows:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t) \left[y_{ij}(t) - x_{ij}(t) \right] + c_2 r_{2j}(t) \left[\hat{y}_j(t) - x_{ij}(t) \right] \quad (2)$$

where $v_{ij}(t)$ describes the velocity of the particle i in the dimension j in the time t, $x_{ij}(t)$ represents the position of the particle in the time t, $y_{ij}(t)$ denotes the best position of the particle, $\hat{y}_j(t)$ describes the global best position of all particles, c_1, c_2 are cognitive and social components, respectively (positive acceleration constants), and $r_{1j}(t), r_{2j}(t)$ are random numbers in the interval $[0, 1]$.

Some research work has been developed to enhance the convergence and the optimal solution obtained with the basic PSO, like the velocity constriction [31], inertia weight [32], and velocity clamping [27].

In this paper, a modification of the gbest PSO variant using the inertia weight (w) is proposed. The inertia weight allows having control in the exploration and exploitation swarm besides the control of the speed and direction of the particles. Equation (2) of the gbest PSO changes as follows:

$$v_{ij}(t+1) = w v_{ij}(t) + c_1 r_{1j}(t) \left[y_{ij}(t) - x_{ij}(t) \right] + c_2 r_{2j}(t) \left[\hat{y}_j(t) - x_{ij}(t) \right] \quad (3)$$

A number of works to dynamically adjust the inertia weight w have been developed, like the random adjustments [33], linear decreasing [26, 34], nonlinear decreasing [34], and fuzzy adaptive inertia [35]. We propose adjusting the inertia weight w and learning factors c_1 and c_2 of the PSO during the iteration process.

The learning factors c_1 and c_2 affect the total velocity of a particle [36]. The cognitive component (c_1) determines the own confidence of the particle and the social component (c_2) defines the confidence of the particle in the neighbors.

We have designed a fuzzy inference system for the adaptation of the inertia weight and learning factors; the design consists of the following:

Two inputs: the number of iterations (NU) when the best fitness does not change and the actual value of the inertia weight (w)

Three outputs: the change in inertia weight (chw) and the change in learning factors c_1 and c_2

Three membership functions for each input and outputs in the fuzzy inference systems: they are shown in Figure 2, defined as LOW, MEDIUM, and HIGH and represented as a left triangle, middle triangle, and right triangle, respectively

Figure 3 illustrates the fuzzy inference system architecture with two inputs (the number of iterations when the best fitness does not change (NU) and the actual inertia weight (w)) and three outputs (the change in inertia weight (chw) and learning factors (c_1) and (c_2)).

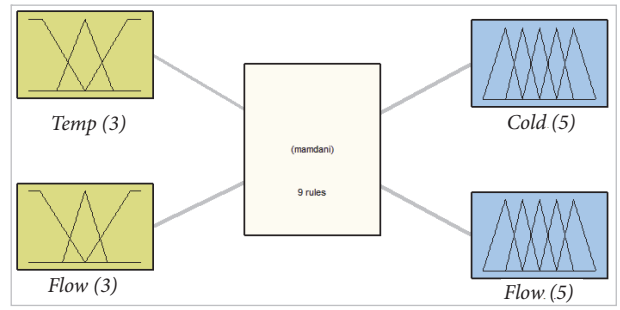

Figure 8: Original representation of the fuzzy system for temperature control.

To obtain the new chw, c_1, and c_2 values, nine fuzzy rules are used in the fuzzy inference system. An example of one fuzzy rule is given as follows:

If NU is MEDIUM, and
w is HIGH
then c_1 is HIGH, c_2 is HIGH and chw is LOW

The rules were obtained by empirical knowledge obtained by experimenting and are shown in Table 1.

The range of NU is in $[1, 10]$, the value for w is in $0.2 \leq w \leq 1.2$, and the values of c_1 and c_2 are in $1.0 \leq c_1, c_2 \leq 2.0$.

The mathematical descriptions for c_1 and c_2 are defined with the following expressions:

$$c_1 = \frac{\sum_{i=1}^{r_{c_1}} \mu_i^{c_1}(c_{1i})}{\sum_{i=1}^{r_{c_1}} \mu_i^{c_1}} \quad (4)$$

where c_1 is percent of cognitive acceleration of the particle i, r_{c_1} is number of fuzzy rules activated to c_1, c_{1i} is output of the fuzzy rule i for c_1, and $\mu_i^{c_1}$ is membership function value of fuzzy rule i for c_1.

$$c_2 = \frac{\sum_{i=1}^{r_{c_2}} \mu_i^{c_2}(c_{2i})}{\sum_{i=1}^{r_{c_2}} \mu_i^{c_2}} \quad (5)$$

where c_2 is percent of cognitive acceleration of the particle i, r_{c_2} is number of fuzzy rules activated to c_2, c_{2i} is output of the fuzzy rule i for c_2, and $\mu_i^{c_2}$ is membership function value of fuzzy rule i for c_2.

For obtaining the values of the variable, the Mamdani FIS model is utilized.

The method of the centroid of area is performed to obtain the defuzzification:

$$O = \frac{\int_z M_o(z)\, z dz}{\int_z M_o(z)\, dz} \quad (6)$$

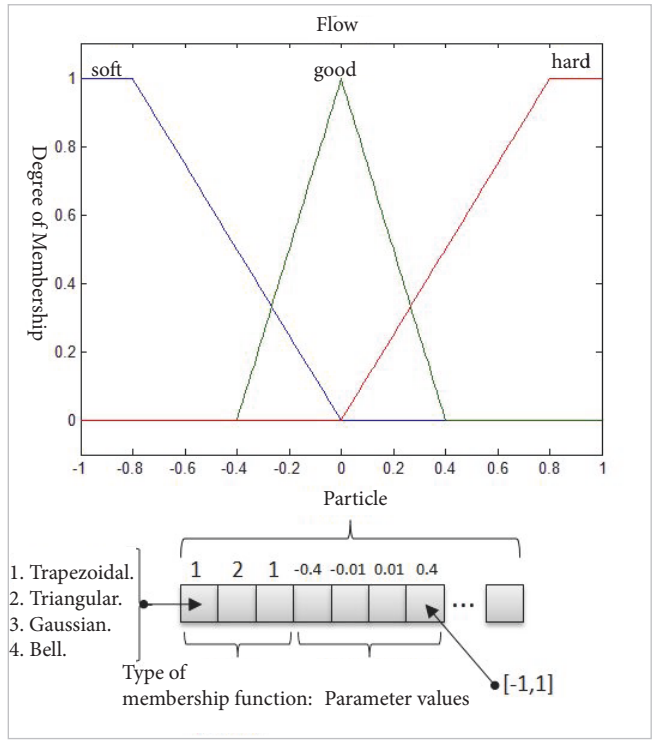

FIGURE 9: Example of the graphical representation of the PSO particles for the membership functions.

3. Ant Colony Optimization

The first algorithm to fall into the framework of the ACO metaheuristics was the Ant System (AS) [2]. In AS, k (artificial) ants build, at the same time, a graph for the solution of the optimization problem. At first, the nodes for the ants are randomly chosen. At each building step, to obtain the node in the next iteration of the ant k, a probabilistic action selection rule, called random proportional rule, is applied [37]. Determining the probability of the ant k to select the path from node i to j is as follows:

$$p_{ij}^{k} = \frac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_i^k}\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}} \tag{7}$$

η_{ij} defines the heuristic value that is independent of the experience of the ant.

α,β are two parameters that define the influence of the pheromone trail and the heuristic information.

N_i^k represents the probable neighborhood of the ant k in the node i, namely, the grouping of nodes that the ant k has not visited yet.

τ_{ij} defines the pheromone evaporation for the route.

The update of the pheromone trails is performed after all the ants build their routes. To reach the update, first the pheromone values on the arcs are reduced by a constant factor and later, on the arcs which the ants passed, pheromone is added. This action is called pheromone evaporation and is performed as follows:

$$\tau_{ij} \longleftarrow (1-\rho)\,\tau_{ij} \tag{8}$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate. The parameter ρ is utilized to elude the accumulation without limits of the pheromone trails and it allows the algorithm to "forget" wrong decisions previously taken. After applying the evaporation, the ants add pheromones on the arcs of their route as follows:

$$\tau_{ij} \longleftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{9}$$

where $\Delta\tau_{ij}^{k}$ is the quantity of pheromones that the ant k provides on the arcs visited. It is determined as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{1}{c^{k}}, & \text{arc}\,(i,j) \in T^{k} \\ 0 & otherwise \end{cases} \tag{10}$$

where c^k represents the longitude of the graph and T^k is calculated as the sum of the longitudes of the arcs of the route for the kth ant.

Many research works dealing with the updating rules for $\Delta\tau_{ij}^{k}$ have been performed, like the Ant System (AS) [2], Elitist Ant System (EAS) [38], Rank-Based Ant System (AS_{rank}) [39], Max-Min ant System (MMAS) [40], and Ant Colony System (ACS) [41].

```
FOR each particle i
   FOR each dimension d
        Initialize position x_id randomly
        Initialize velocity v_id randomly
   END FOR
END FOR
Iteration k=1
DO
   FOR each particle i
          Calculate fitness value
          IF the fitness value is better than p_best_id in history
             Assign current fitness value as the p_best_id
          END IF
          IF the fitness value is better than g_best_id in history
             Assign current fitness value as the g_best_id
          END IF
   END FOR
   FOR each particle i
        FOR each dimension d
                Calculate w, c1 and c2 with the fuzzy inference system
                Calculate velocity using the equation
                v_id (k+1) = w v_id (k) + c1 Rand_1 (p_id - x_id) + c2 Rand2 (p_gd - x_id)
                   Update particle position using the equation
                   x_id (k+1) = x_id (k) + v_id (k+1)
        END FOR
   END FOR
k = k + 1
WHILE Maximum iterations or mínimum error criteria are not attained
```

Algorithm 1: PSO algorithm.

```
   FOR each ant i
      Initialize ant i
   END FOR
   Iteration k=1
   DO
      FOR each particle i
         Construct the trail
         Select the next node
         Calculate the value with the fuzzy inference system
         Update the local pheromone trail
         IF the trail is complete
            Calculate the value with the fuzzy inference system
            Update the global pheromone trail
         ELSE
            GO to Select the next node
         END IF
      END FOR
      k = k + 1
   WHILE Maximum iterations or mínimum error criteria are not attained
```

Algorithm 2: ACO algorithm.

In this paper, we used an Ant Colony System (ACS) [41] for three important reasons. At first instance, the ACS uses an action choice rule that is more aggressive than the AS in the exploitation of the search experience concentrated by the strongest ants. Second, the ACS applies the pheromone deposit and the pheromone evaporation only in the arcs of the more promising route. Third, when the ant moves from node i to node j using the arc (i, j), with the objective of incrementing the exploration, it eliminates a small amount of pheromone from the arc [42, 43].

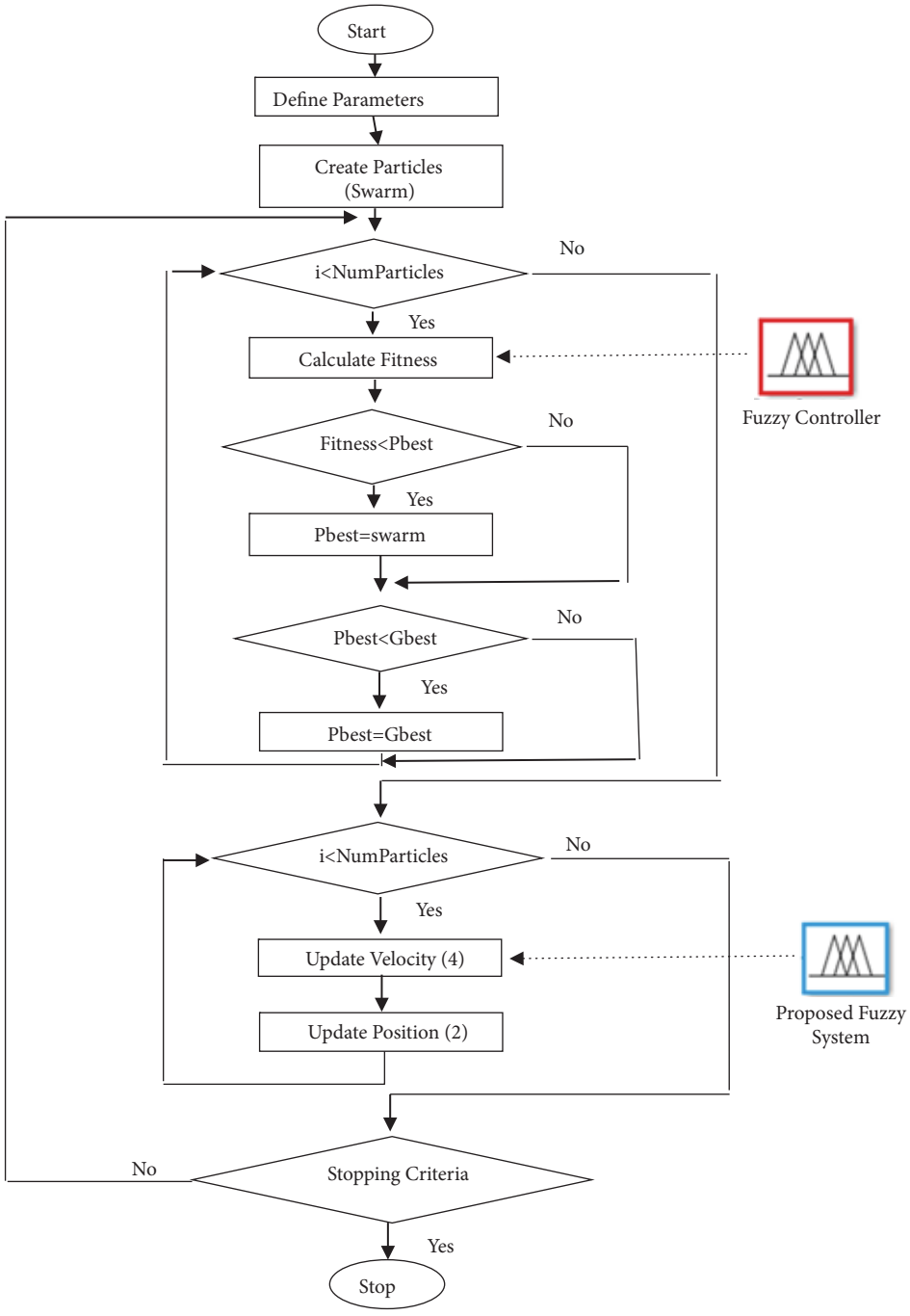

FIGURE 10: The flowchart of PSO.

ACO has a process in which it does not use heuristic values and the part of the consequent can be merely selected as follows:

$$p_{ij}^{k} = \frac{\left[\tau_{ij}\right]^{\alpha}}{\sum_{l \in N_i^k} \left[\tau_{ij}\right]^{\alpha}} \tag{11}$$

In ACS, the ant k moves from node i to node j according to a pseudo-random-proportional rule as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} \underset{j \in N_i^k [\tau_{ij}]^{\alpha}}{\text{argmax}}, & if\ q \le q_0 \\ J & otherwise \end{cases} \tag{12}$$

where q is random value uniformly distributed in $[0, 1]$, q_0 is parameter in the interval $[0, 1]$, and J is random value calculated by (11) (with $\alpha = 1$).

In the global pheromone trial update for ACS, in each iteration, only the global best ant has authorization to add the

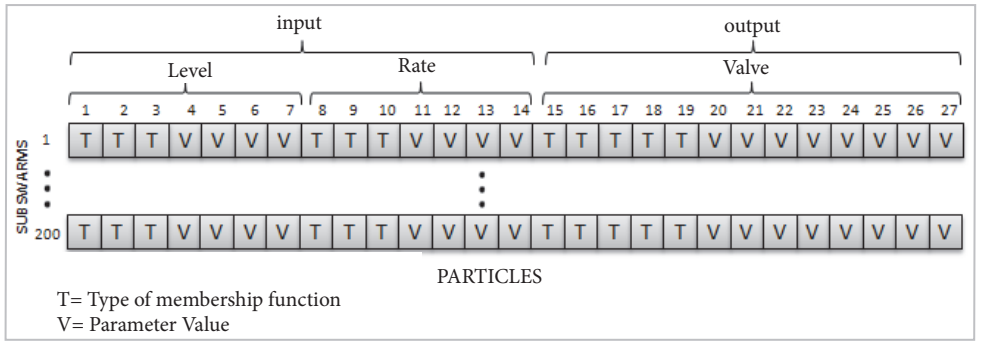

Figure 11: Representation of the PSO for the optimization of membership functions.

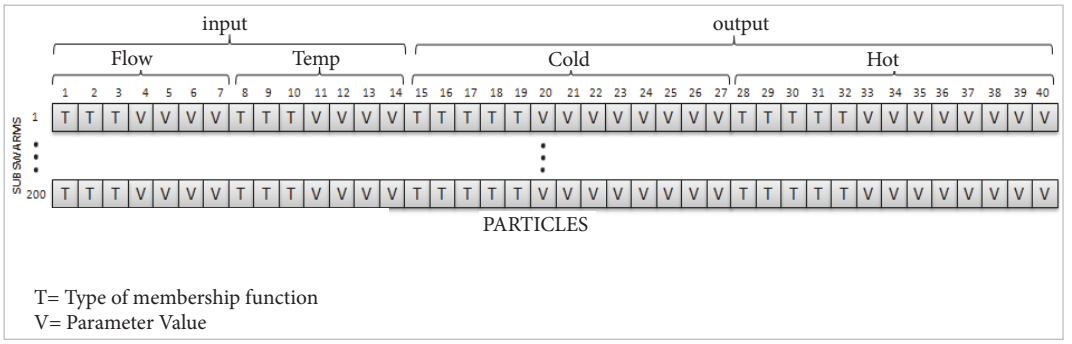

Figure 12: Representation of PSO for the optimization of membership functions.

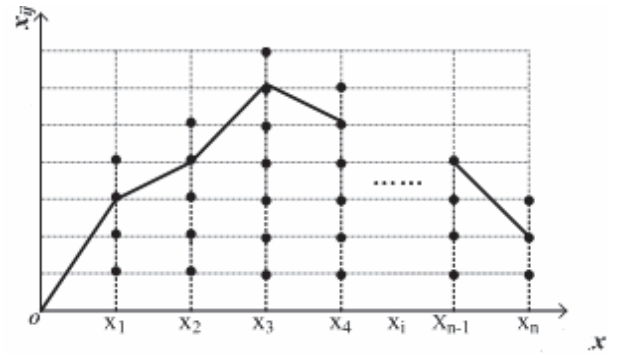

Figure 13: Variable division and ant search path.

pheromone. Accordingly, the update in ACS is performed as follows:

$$\tau_{ij} \longleftarrow (1-\rho)\,\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \tag{13}$$

where $\Delta\tau_{ij}^{bs} = 1/C^{bs}$; in ACS, the pheromone trail update is applied to the arcs of τ^{bs}, not to all the arcs as in AS.

We have defined a fuzzy inference system for pheromone evaporation ρ which is made up of the following elements:

Two inputs: the value of $\Delta\tau_{ij}^{bs}$ and the pheromone trail τ_{ij}

One output: the change in pheromone evaporation ρ

Three membership functions for each input and output: LOW, MEDIUM, and HIGH, respectively, represented as membership functions of left triangle, middle triangle, and right triangle. Figure 4 illustrates the fuzzy system to obtain the change in pheromone evaporation ρ

Nine fuzzy rules are considered from which the changes in ρ are calculated. The rules are obtained by experimental knowledge as shown in Table 2

The ranges of $\Delta\tau^{bs}$ and τ are normalized into $[0, 1]$

In ACS, besides the global pheromone trail update rule, a local pheromone update rule is applied after the ants cross an arc (i, j) in the construction of the graph. This rule is given by the following equation:

$$\tau_{ij} \longleftarrow (1-\xi)\,\tau_{ij} + \xi\tau_0 \tag{14}$$

where $\xi, 0 < \xi < 1$. The value of τ_0 is configured to take the initial value of the pheromone trails.

We have defined a fuzzy inference system for pheromone evaporation ξ which consists of the following components:

Two inputs: the maximum probability of choosing the next node p_{ij}^k and the pheromone trail τ_{ij}

One output: the change in pheromone evaporation ξ

Three membership functions for each input and output: LOW, MEDIUM, and HIGH, respectively, represented as membership functions of left triangle,

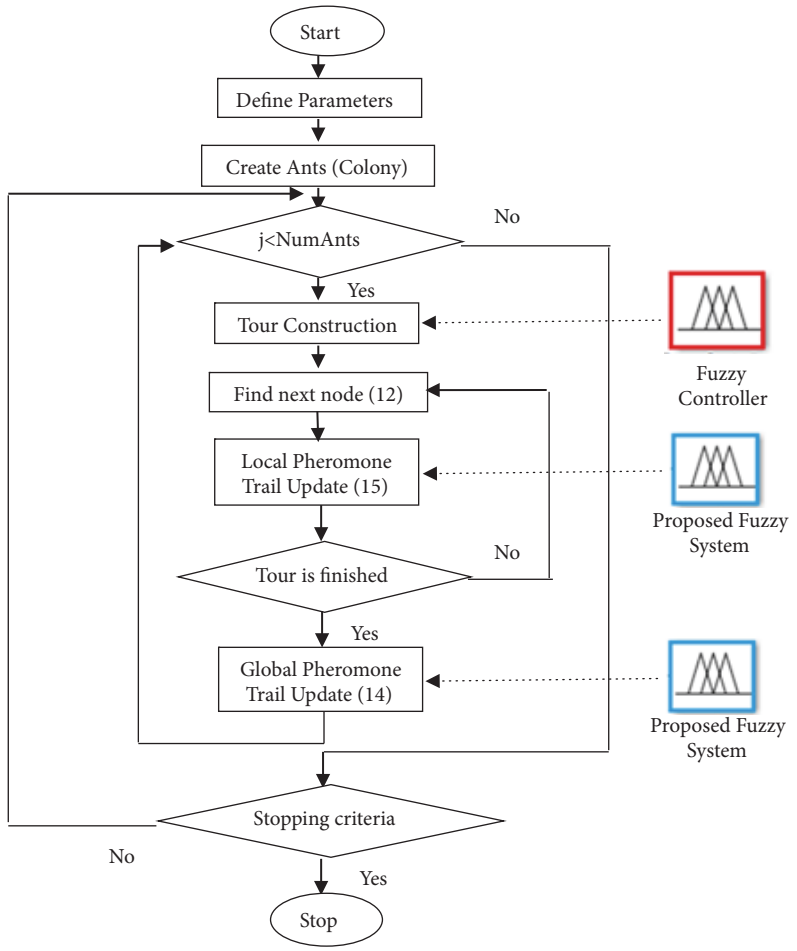

Figure 14: The flowchart of the ACO algorithm.

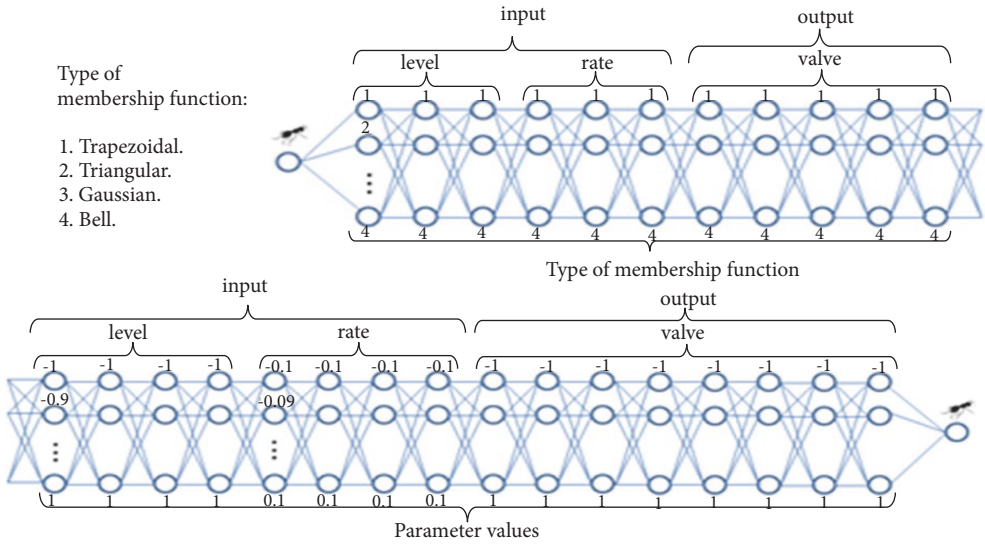

Figure 15: Representation of the potential solutions in the pheromone matrix for the water system.

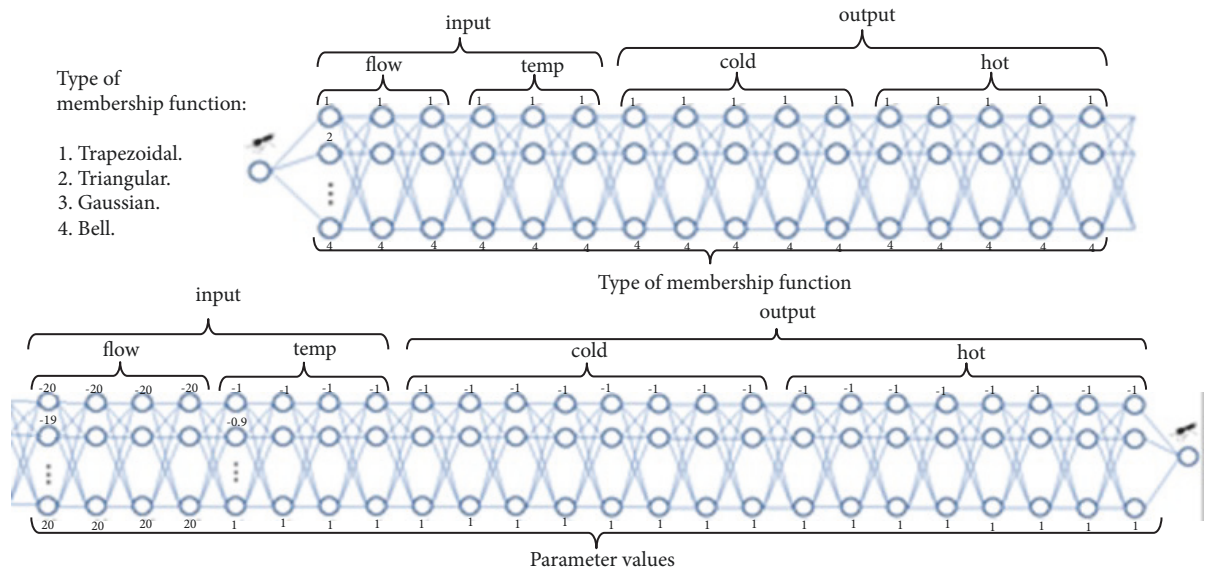

Figure 16: Representation of the potential solutions in the pheromone matrix for temperature system.

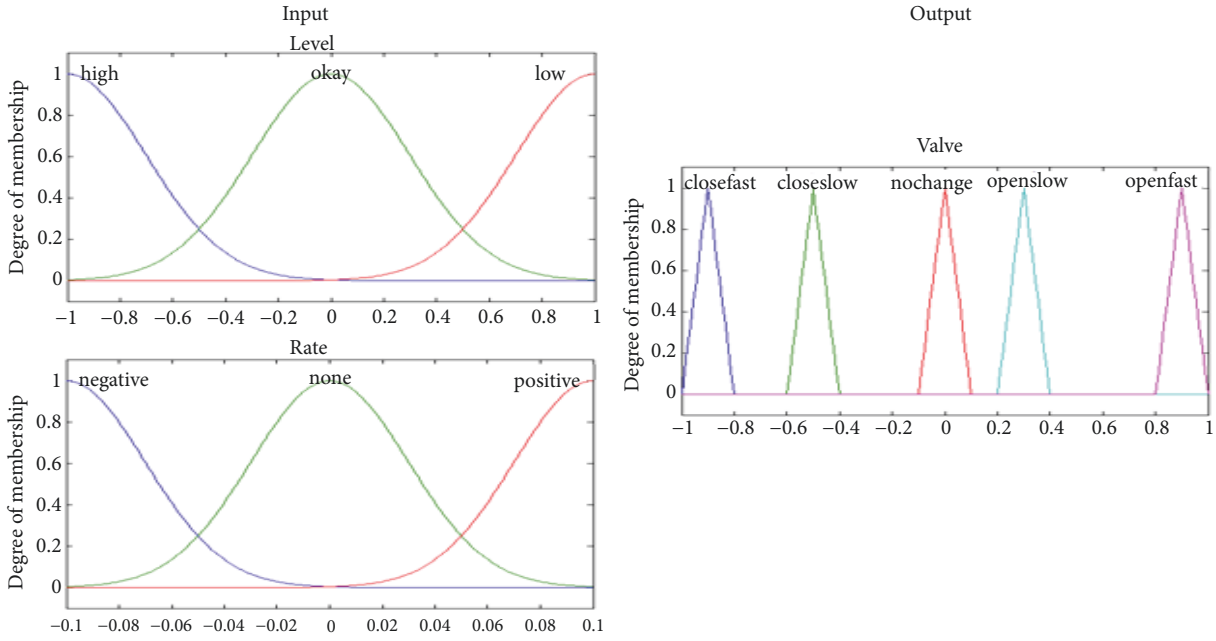

Figure 17: Membership functions for the water tank FLC without optimization.

middle triangle, and right triangle. Figure 5 illustrates the fuzzy inference system to obtain the change in pheromone evaporation ξ

Nine fuzzy rules are considered from which the changes in ξ are calculated. The rules are obtained by experimental knowledge as shown in Table 3

The ranges of p and τ are normalized into [0, 1]

3.1. Water Level Control in a Tank. The water tank model and the control of this water tank using a fuzzy control system is the system used for the experiments of the proposed approach [22]. For this case, the control of the water that flows into the tank is made by using one valve. The outflow rate depends on the diameter of the output pipe, which is constant, and the pressure in the tank, which varies with the water level. Therefore, the system has nonlinear characteristics. The main objective of the fuzzy logic controller (FLC) for the water tank is to maintain the water in between the lower level L and upper level U according to the reference R, as shown in Figure 6. The FLC performs the control of the level and velocity to the water, giving as output of the FLC the quickness of flow out of the pipe.

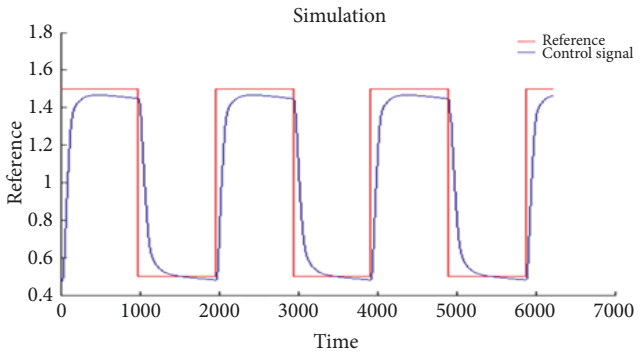

Figure 18: Behavior of the water level in the tank with the FLC not optimized.

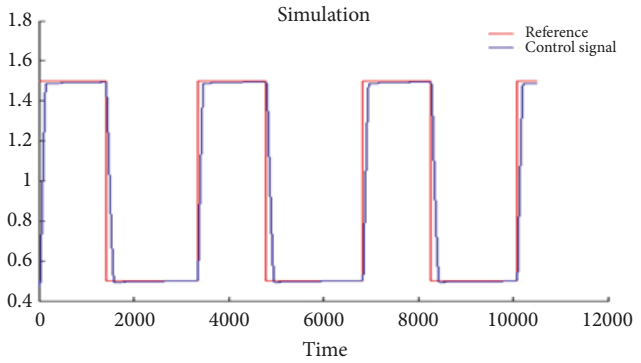

Figure 19: Behavior for the optimized water tank FLC with PSO.

The controller needs to be efficient to manage flow in the pipe and maintain the current level of water in the tank. The water level error is the input for the controller (the difference between the desired water level and the current water level) and the rate of opening or closing for the valve is the output.

As Figure 7 illustrates, the fuzzy controller has two inputs: the rate of change of the water level and the difference between the actual and the required water levels. The controller uses these inputs to manipulate the inflow rate.

For the fuzzy system of the water tank, 11 membership functions were used: 3 per each input and 5 at the output and all this is illustrated in Figure 7. The linguistic variable *level* is defined to have three fuzzy sets, HIGH, OK, and LOW, with the associated membership functions as *leftGaussian*, *Gaussian, and rightGaussian,* respectively; the linguistic variable *rate* is defined to have three fuzzy sets, NEGATIVE, NONE, and POSITIVE, with associated membership functions as *leftGaussian*, *Gaussian, and rigthGaussian,* respectively; the linguistic variable *valve* is defined to have five fuzzy sets, CLOSE_FAST, CLOSE_SLOW, NO_CHANGE, OPEN_SLOW, and OPEN_FAST, with associated membership function as *leftTriangle*, *Triangle, and rigthTriangle,* respectively. The following five fuzzy rules are used to calculate the change in the valve:

(1) If (level is OKAY) then (valve in NO_CHANGE)

(2) If (level is LOW) then (valve in OPEN_FAST)

(3) If (level is HIGH) then (valve in CLOSE_FAST)

(4) If (level is OKAY) and (rate is POSITIVE) then (valve in CLOSE_SLOW)

(5) If (level is OKAY) and (rate is NEGATIVE) then (valve in OPEN_SLOW)

3.2. Temperature Control in a Shower. We also used as basis for our experiments the problem of temperature control and the fuzzy system presented in [22]. As Figure 8 illustrates, the fuzzy controller consists of two inputs: the water temperature and the flow velocity. The controller utilizes the outputs obtained with the two inputs to control the cold and hot valves.

For the fuzzy system of the temperature control, 16 membership functions were used: 3 per each input and 5 at the output and all this is illustrated in Figure 8. The fuzzy variable *temp* is defined to have three fuzzy sets, COLD, GOOD, and HOT, with associated membership functions as *leftTriangle*, *Triangle, and rigthTriangle,* respectively; the fuzzy variable *flow* is defined to have three fuzzy sets, SOFT, GOOD, and HARD, with associated membership function as *leftTriangle*, *Triangle, and rigthTriangle,* respectively; the fuzzy variables *cold* and *hot* are defined to have five fuzzy sets, CLOSEFAST, CLOSESLOW, STEADY, OPENSLOW, and OPENFAST, with associated membership functions as *leftTriangle*, *Triangle, and rigthTriangle,* respectively. The following nine fuzzy rules were used to calculate the change of the flow and temperature in the outputs:

(1) If (temp is COLD) and (flow is SOFT) then (cold is OPENSLOW) and (hot is OPENFAST)

(2) If (temp is COLD) and (flow is GOOD) then (cold is CLOSESLOW) and (hot is OPENSLOW)

(3) If (temp is COLD) and (flow is HARD) then (cold is CLOSEFAST) and (hot is CLOSESLOW)

(4) If (temp is GOOD) and (flow is SOFT) then (cold is OPENSLOW) and (hot is OPENSLOW)

(5) If (temp is GOOD) and (flow is GOOD) then (cold is STEADY) and (hot is STEADY)

(6) If (temp is GOOD) and (flow is HARD) then (cold is CLOSESLOW) and (hot is CLOSESLOW)

(7) If (temp is HOT) and (flow is SOFT) then (cold is OPENFAST) and (hot is OPENSLOW)

(8) If (temp is HOT) and (flow is GOOD) then (cold is OPENSLOW) and (hot is CLOSESLOW)

(9) If (temp is HOT) and (flow is HARD) then (cold is CLOSESLOW) and (hot is CLOSEFAST)

3.2.1. Implementation of the Proposed Methodology. In this section, the PSO and ACO algorithms are implemented for the optimization of two FLC systems [44, 45]. The optimization approach has been designed in the following sequence: first, the optimization defines the type of membership function: can be Gaussian, triangular, trapezoidal, or generalized bell; later, the optimization determines the parameters for the chosen membership functions; it is for the interesting values depending on the type of membership

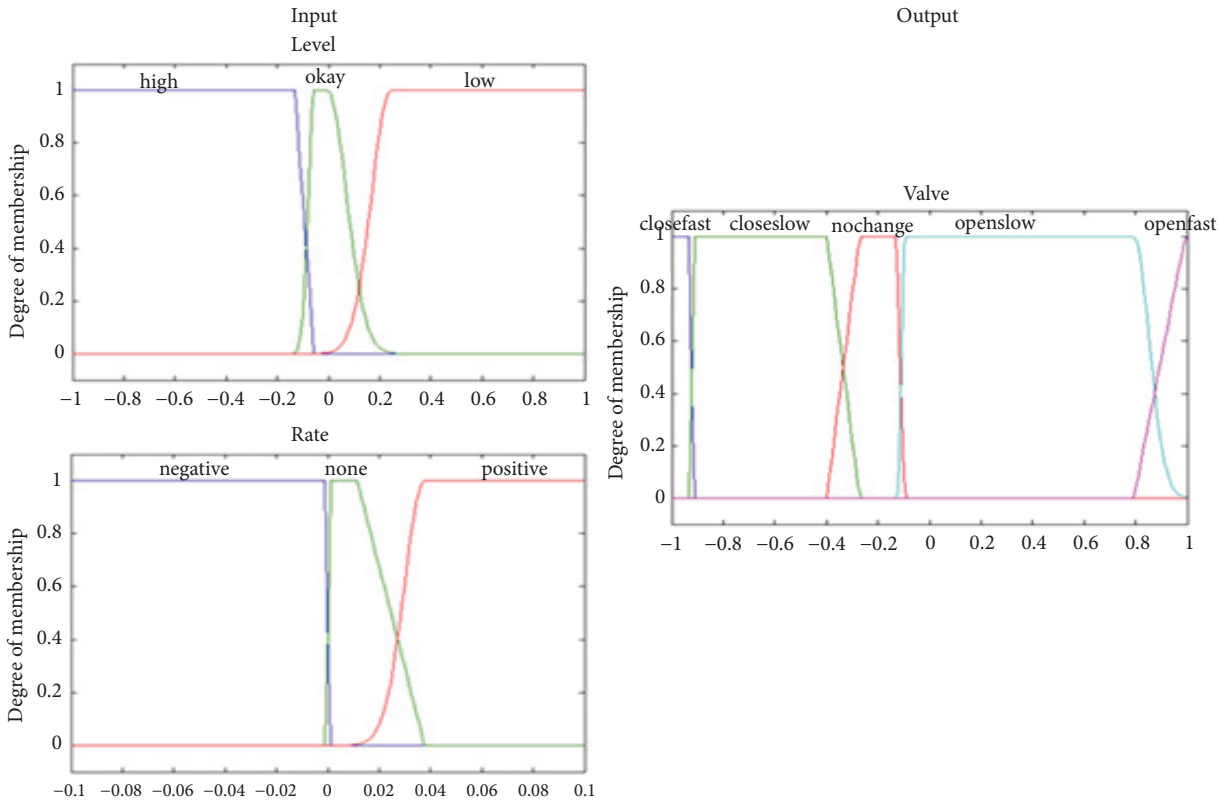

Figure 20: Membership functions optimized with PSO for the water tank FLC.

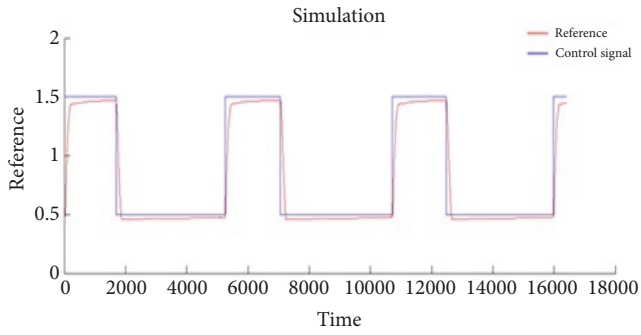

Figure 21: Behavior for the optimized water tank FLC with ACO.

function, like the coordinates of the points of edge, the standard deviation, and so forth.

The objective function is built to represent the fitness of a solution and is considered as an interface between optimization problem and the algorithm. In order to determine the fitness of the fuzzy model, the mean square error (MSE) is used (defined in (15)). The goal error for MSE is zero.

$$MSE = \frac{1}{N}\sum_{k=1}^{n}\left[y(k) - \tilde{y}(k)\right]^2 \quad (15)$$

where $y(k)$ is desired output, $\tilde{y}(k)$ is calculated output of the proposed methodology, and N is number of pieces of data by the validation of model.

3.3. Proposed Methodology for PSO. In this case, we represented the problem as is shown in Figure 9, where one particle can be divided into sections to represent the type and parameters of the membership function.

PSO is used to optimize fuzzy controllers for two cases. In one, PSO is used to optimize the FLC from the water control in a tank and in the other PSO is used to optimize the temperature control in a shower. They both used a set of 200 particles and 300 iterations and they perform 30 experiments. We use 200 particles and 300 iterations because after making several tests the best results were obtained with these values. In Section 3.6.3, we show a statistical test with the 30 experiments against ACO to validate in the best way the experimental results. The flowchart and pseudocode of the PSO algorithm are displayed in Figure 10 and Algorithm 1 of PSO, respectively, and the process is described as follows:

(1) In the search space of the problem, we use random values for the positions and velocities of the particles; this is to create and initialize the population.

(2) The fitness (in the fuzzy controller) is evaluated for all particles using the established objective function, MSE.

(3) A comparison between the current fitness against the previous fitness of each particle is applied. If the current fitness is better than the previous fitness, then the current value is established for the *pbest* and the current position like the *pbest* position.

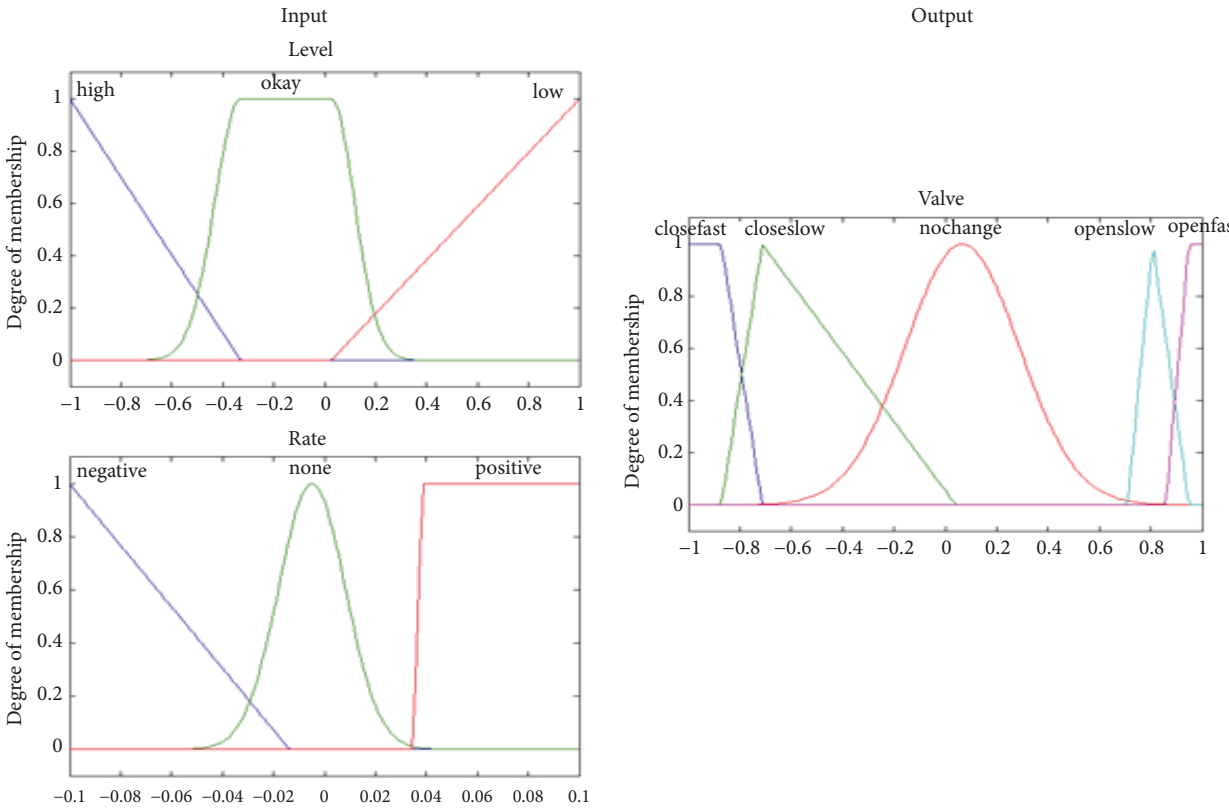

Figure 22: Membership functions optimized for the water tank FLC with ACO.

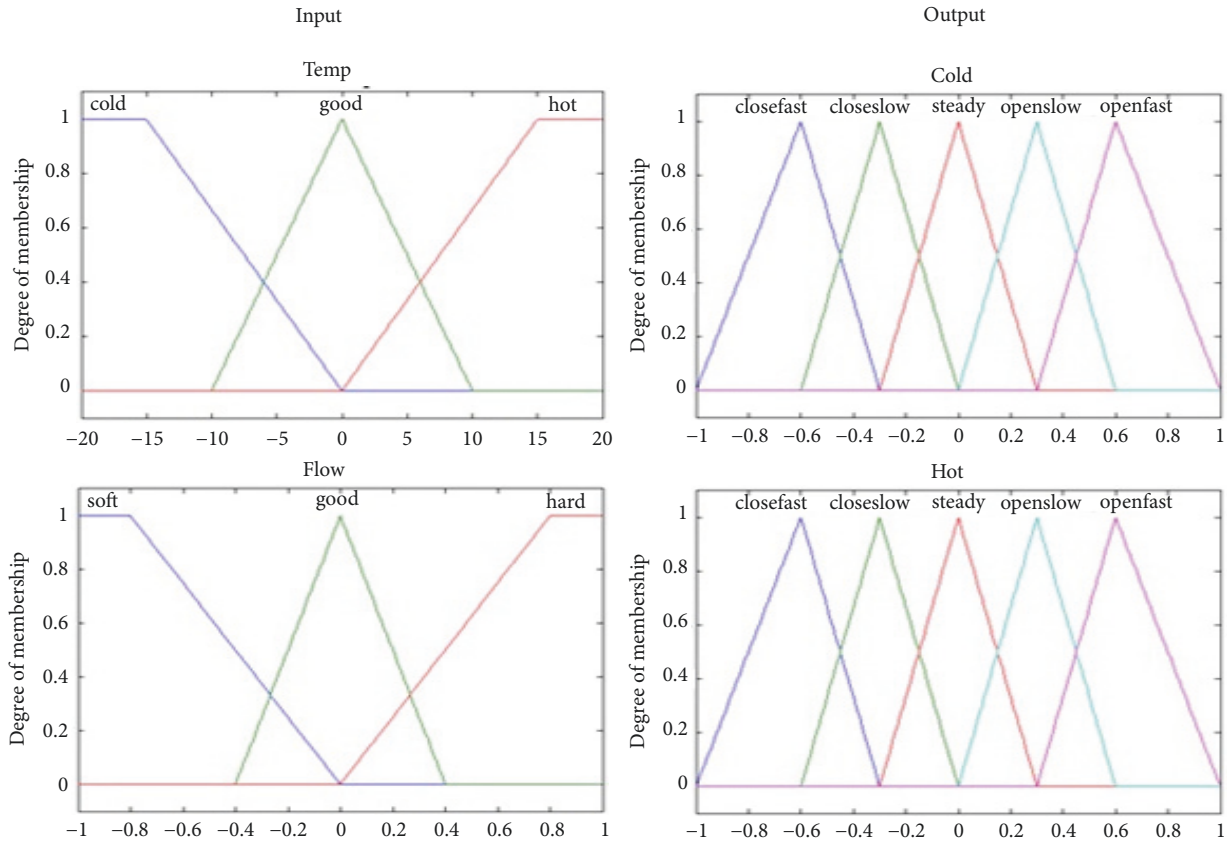

Figure 23: Membership functions not optimized for the temperature control FLC.

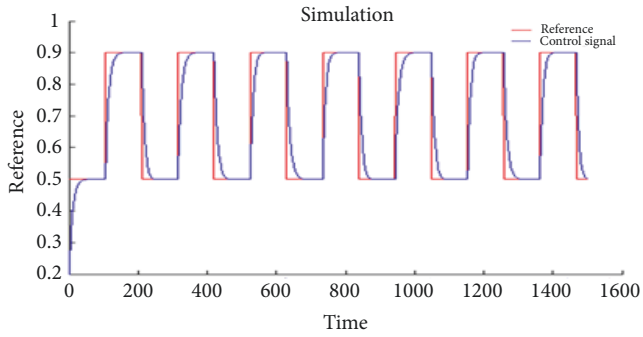

Figure 24: Behavior of the not optimized temperature control in the FLC for the shower.

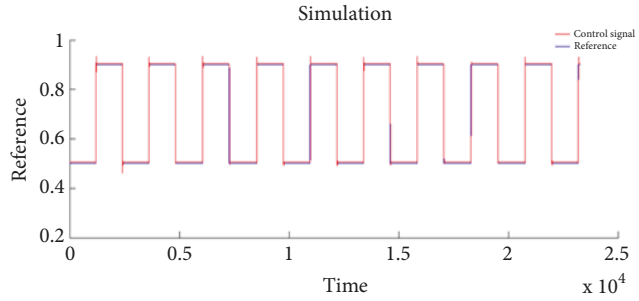

Figure 25: Behavior for the optimized temperature control FLC with PSO.

(4) A comparison among the current fitness of each particle against the best global fitness (*gbest*) of the previous population is applied. If the current fitness is better than *gbest*, then the index and value of the current particle are established like the *gbest*.

(5) The evaluation is performed with the star topology:

(5.1) In base to (3) and the proposed fuzzy inference system (Figure 3), the particle velocity is calculated.

(5.2) Actualize the particle position using (1).

(6) Steps (2) to (6) are repeated until the algorithm reaches the maximum number of iterations or the best fitness is better than the desired value.

3.3.1. PSO for Water Level Control. Figure 11 shows the graphical representation of the particles used for the optimization of the FLC for tank control. As it was mentioned before, we have implemented a set of subswarms of size of 27 particles, where we can see that each subswarm is formed with two value types, such as the type of membership function and parameter value corresponding to each variable (input-output).

3.3.2. PSO for Temperature Control. Figure 12 shows the graphical representation of particles used for the optimization of the FLC from tank control. As mentioned before, we have implemented a set of subswarms of size of 40 particles, where we can see that each subswarm is formed with two value types such as type of membership function and parameter value corresponding to each variable (input-output).

3.4. Proposed Methodology for ACO. In the Ant Colony Optimization algorithm, the problem to optimize must be represented as a combinatorial problem; in this case, the types of membership functions should be represented as a graph; it means that the potential values require to be represented as a graph. This is an important part, for the reason that ACO operates with this type of representation. Each node must be a possible solution that an ant can take.

The ant can consider each node like a potential solution to choose. At the end of the iterations, every ant will have a graph with the potential values to make the evaluation; in this graph, if we suppose that the problem to optimize is used to design a graph with q_i nodes, the node value is determined as x_{ij} with i as the variable index (i=1, 2,..., n) and j as the number node for x_i ($j = 1, 2, q_{ij}$). After the division, a network that is composed of variables and their division is constructed as depicted in Figure 13.

The flowchart for ACO is displayed in Figure 14, and the illustration of the ACO algorithm is shown in Algorithm 2 and the process is described as follows:

(1) The ant colony is created and initialized with random positions. We set the initial parameters for ACO making several trial and error experiments. For each ant, construct a graph (for the fuzzy controller).

(1.1) The next node is selected in function to the probability obtained with (11).

(1.2) Reduce the local pheromone trail (evaporation) according to (14) with the proposed fuzzy system (Figure 7).

(2) Reduce the global pheromone trail (evaporation) according to (14) with the proposed fuzzy system (Figure 6).

(3) Steps (2) to (4) are repeated until the algorithm reaches the maximum iterations or the best fitness is better than the desired value.

3.4.1. Water Level Control in a Tank. In this paper, the design of fuzzy logic control (FLC) is performed using Ant Colony System. Figure 15 shows the graphical representation of the potential solutions in the pheromone matrix used for the optimization of FLC from tank control, where the first node of the graph represents the nest and the food source is the last node. This implementation is structured with a total of 27 nodes, where each node is formed for a type of membership function or parameter value (input-output).

3.4.2. Temperature Control in a Shower. We also used the Ant Colony System algorithm for this case. Figure 16 shows the graphical representation of the potential solutions in the pheromone matrix used for the optimization of FLC from

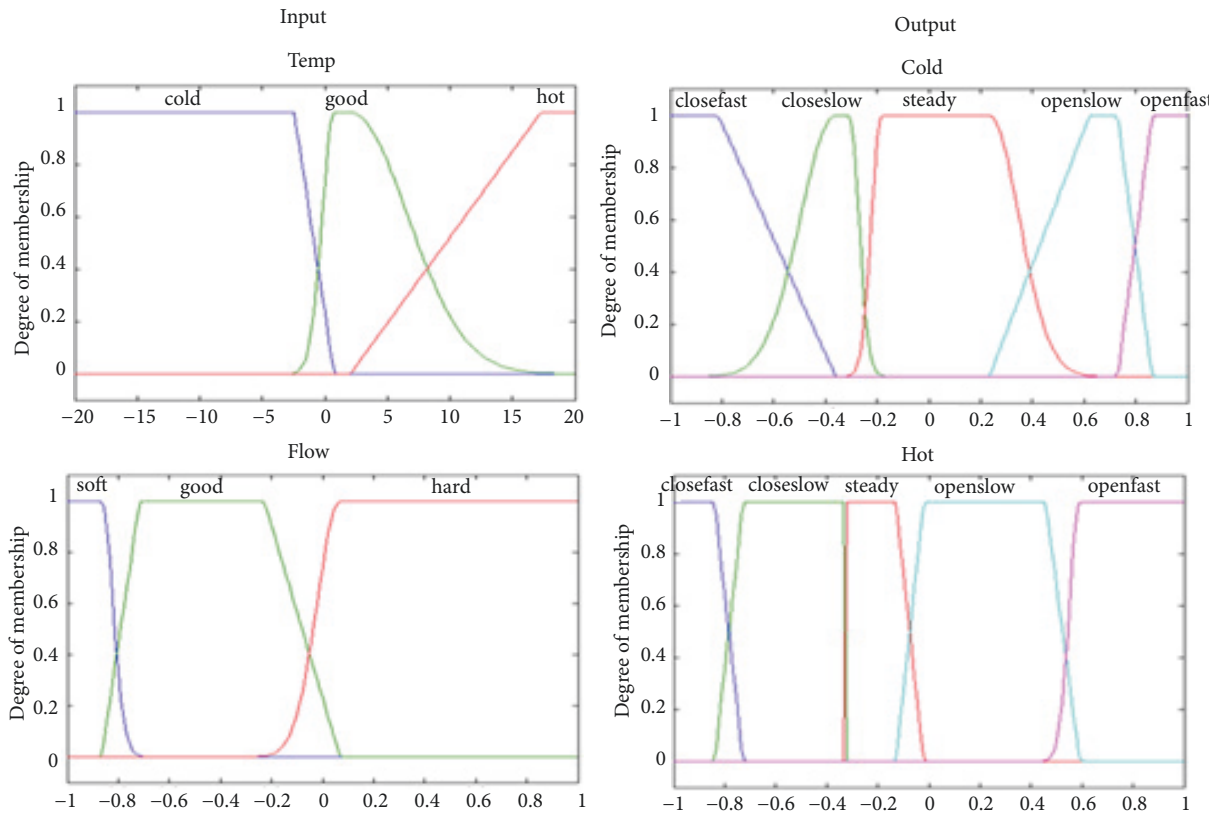

Figure 26: Membership functions optimized for the temperature control FLC with PSO.

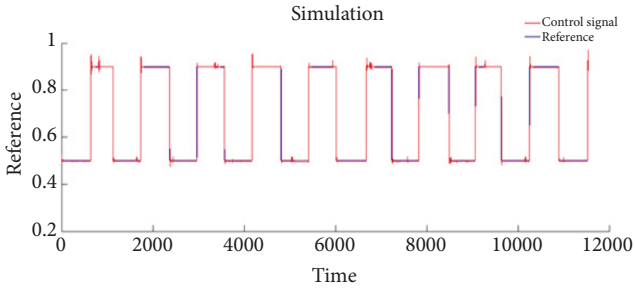

Figure 27: Behavior for the optimized temperature control FLC with ACO.

tank control, where the first node of the graph represents the nest and the food source is the last node. This implementation is structured with a total of 40 nodes, where each node is formed for a type of membership function or parameter value (input-output).

3.5. Simulation Results. In this part, the results of the optimization of the FLCs for the water level control and temperature control systems with Particle Swarm Optimization and Ant Colony Optimization are presented. For comparison purposes, we used FLCs provided by [22] which have not been optimized.

3.5.1. Simulation Results for the Water Level Control. Figure 17 shows the fuzzy logic control (FLC) for the water tank without optimization, the type of membership functions, and their structure.

Figure 18 shows the original simulation in relation with the Mean Square Error (described in Section 3) obtaining an error of 0.0668 when the fuzzy system was not optimized, where $\tilde{y}$ is taken as the control signal, y as the reference signal, and N as the number of data points.

3.5.2. Simulation Results for Water Level Control with PSO. The best experiments, with the smaller error of 0.026952, are presented in Figure 19; Figure 19 shows the behavior of the reference signal against the control signal in simulation for the optimization with PSO; the resulting new structures of membership functions after parameter optimization for the water tank FLC are shown in Figure 20.

3.5.3. Simulation Results for Water Level Control with ACO. The best experiments, with a smaller error of 0.045792, are presented in Figure 21; Figure 21 shows the behavior of the reference signal against the control signal in simulation for the optimization with ACO; the resulting new structures of membership functions after parameter optimization for the water tank FLC are shown in Figure 22.

3.6. Simulation Results for Temperature Control. Figure 23 shows the fuzzy logic control (FLC) for the shower temperature without optimization, the type of membership functions, and their structure.

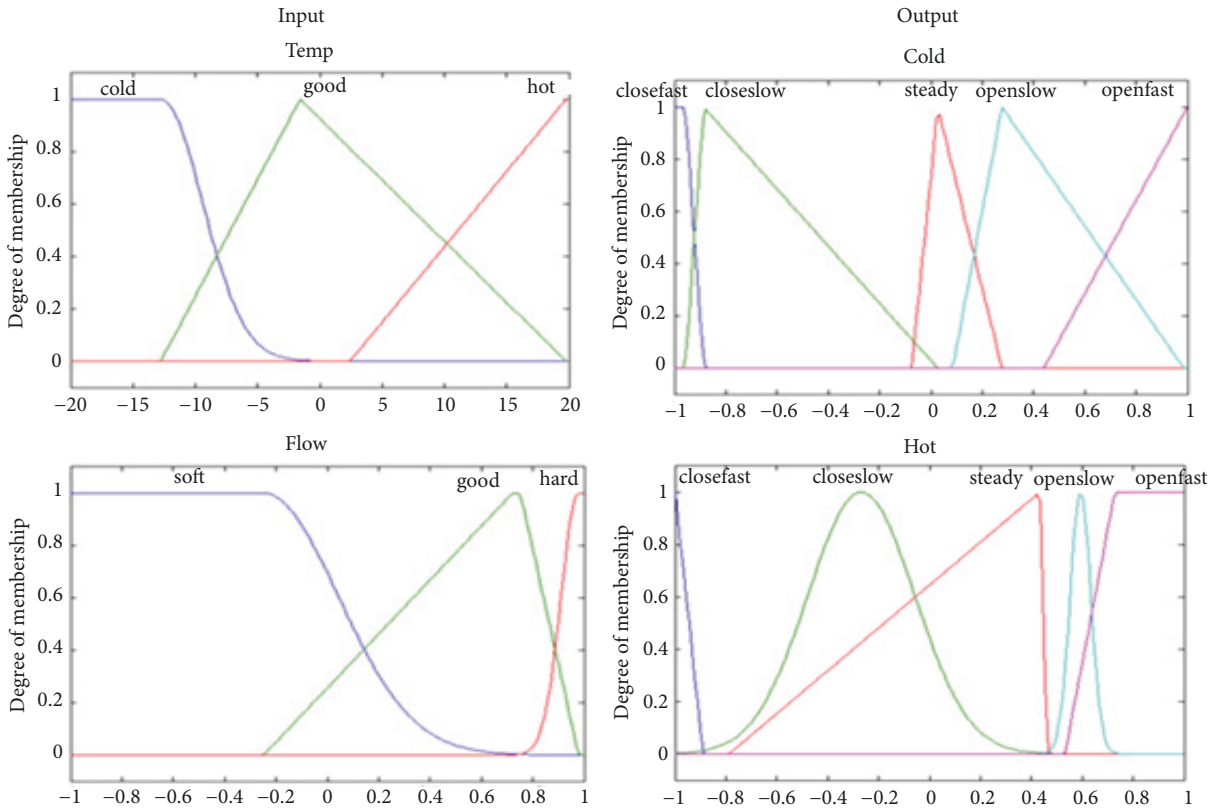

Figure 28: Membership functions optimized for the temperature control FLC with ACO.

Figure 24 shows the original simulation in relation with the Mean Square Error (described in Section 3) obtaining an error of 0.007013 when the fuzzy system was not optimized, where $\tilde{y}$ is taken as the control signal, y as the reference signal, and N as the number of data points.

3.6.1. Simulation Results for Temperature Control with PSO. The best experiment, with a smaller error of 0.0003758, is presented in Figure 25; Figure 25 shows the behavior of the reference signal against the control signal in simulation for the optimization with PSO; the resulting new structures of membership functions after parameter optimization for the shower temperature FLC are shown in Figure 26.

3.6.2. Simulation Results for Temperature Control with ACO. The best experiment, with a smaller error of 0.00076356, is presented in Figure 27; Figure 27 shows the behavior of the reference signal against the control signal in simulation for the optimization with ACO; the resulting new structures of membership functions after parameter optimization for the shower temperature FLC are shown in Figure 28.

3.6.3. Comparative Study of Optimized ACO versus Optimized PSO: The Case of Temperature Control in a Shower. In this section, we included a statistical test to validate in the best way our approach between the ACO and PSO optimized methods. We made 100 experiments for the case of the temperature control with both methods. The following table shows the results with a Student's t-test. We use 30 samples, selected randomly from the 100 experiments to make this test. We can see in Table 4 for this case that the ACO method was better than PSO because the T-value was -7.28; however, in Table 5, we can see that the 30 experiments and the values are very similar with both methods; however, some values of PSO are larger than 1, and for the ACO algorithm, the values are smaller than 1. Therefore, the differences for these two methods using optimized controllers were significant.

Table 4: Student's t-test.

N	Mean	Standard deviation	SE mean
30	0.190	0.163	0.030
30	0.807	0.434	0.079

Difference = μ (ACO) - μ (PSO).
Estimate for difference: -0.6164.
95% lower bound for difference: -0.7593.
T-value = -7.28
P value = $1.20587E-08$.

4. Conclusions

In this work, we have applied the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) metaheuristics for the design of fuzzy systems. For this purpose, we have adopted one ACO variant, namely, the Ant Colony System (ACS). Both algorithms, ACS and PSO, have changing

Table 5: Experimental results with ACO and PSO.

Experiment	PSO	ACO
1	1.1456	0.0067823
2	1.2076	0.012086
3	0.90314	0.0281
4	0.42635	0.033326
5	0.20136	0.051824
6	0.75873	0.066958
7	0.093008	0.077077
8	0.68108	0.080195
9	1.1802	0.083176
10	1.1365	0.10675
11	1.1563	0.10827
12	0.31743	0.12463
13	1.2254	0.13095
14	1.0505	0.13263
15	1.2925	0.13276
16	0.40378	0.14014
17	1.2477	0.14722
18	1.1119	0.16984
19	0.63035	0.17191
20	1.0559	0.17539
21	0.41091	0.22107
22	0.069772	0.22883
23	1.2036	0.23622
24	1.1208	0.3004
25	0.078089	0.30398
26	1.071	0.30562
27	0.30528	0.4007
28	1.2911	0.49584
29	1.1934	0.57976
30	0.23343	0.65851
Error average	0.806756967	0.19036481

parameters with fuzzy dynamic adaptation during execution. For the ACS algorithm, the global and local evaporation of the pheromone is dynamically adapted using a fuzzy system. For the PSO algorithm, the inertia weight and the parameters c_1 and c_2 are adapted dynamically with a fuzzy system. The type of membership functions and the parameters of membership function of the fuzzy logic controllers for temperature control and water level control were optimized with ACO and PSO algorithms, respectively. The results of both optimization algorithms are compared. Those results show good solutions in comparison with the original problem that is not optimized.

The use of metaheuristic algorithms is amply utilized in optimization of problems; for the benchmark problems of temperature control and water level control, the implemented fuzzy systems are used to adjust parameters of the ACO and PSO and improve the results of the algorithms.

As a general conclusion with the statistical test applied with both analyzed methods, we can demonstrate that, with the dynamic parameter adaptation, ACO and PSO can be good alternatives to solve some cases of control; the experimental results are very similar to the two optimized methods; however, for the case of temperature control in a shower, ACO was slightly superior to PSO. Also, there are other experiments to validate this approach with other cases, for example, the water tank control, and in some cases PSO is better than ACO.

For future work, we plan to hybridize these algorithms to compare results and test with fuzzy inference systems for different membership functions or implement type 2 fuzzy inference systems to enhance the results of the work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest of any kind regarding the publication of this paper.

Acknowledgments

The authors would like to express their gratitude to CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

[1] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publisher, 2001.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

[4] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micromachine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.

[5] D. S. Johnson and L. A. McGeoch, "The traveling salesman problem: a case study," in *Local search in combinatorial optimization*, Wiley-Intersci. Ser. Discrete Math. Optim., pp. 215–310, Wiley, Chichester, 1997.

[6] G. Reinelt, *The Traveling Salesman Problem: Computational Solutions for TSP Applications*, vol. 840 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 1994.

[7] R. Martínez-Marroquín, O. Castillo, and J. Soria, "Parameter tuning of membership functions of a fuzzy logic controller for an autonomous wheeled mobile robot using ant colony optimization," in *Proceedings of the 2009 IEEE International Conference on Fuzzy Systems*, pp. 2007–2012, kor, August 2009.

[8] E. Lizarraga, O. Castillo, J. Soria, and F. Valdez, "A Fuzzy Control Design for an Autonomous Mobile Robot Using Ant Colony

Optimization," in *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, vol. 547 of *Studies in Computational Intelligence*, pp. 289–304, Springer International Publishing, Cham, 2014.

[9] P. Kaur, Sh. Kumar, and A. Singh, "Optimization of Membership Functions Based on Ant Colony Algorithm," *International Journal of Computer Science and Information Security*, vol. 10, 2010.

[10] H. Jiang, H. Deng, and Y. He, "Determination of fuzzy logic membership functions using extended ant colony optimization algorithm," in *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008*, pp. 581–585, chn, October 2008.

[11] F. Valdez, P. Melin, and O. Castillo, "Evolutionary method combining particle swarm optimization and genetic algorithms using fuzzy logic for decision making," in *Proceedings of the 2009 IEEE International Conference on Fuzzy Systems*, pp. 2114–2119, kor, August 2009.

[12] P. Angeline, "Evolutionary optimization versus particle swarm optimization: philosophy and performance differences," in *Evolutionary Programming VII*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., vol. 1447 of *Lecture Notes in Computer Science*, pp. 601–610, Springer, Berlin, Germany, 1998.

[13] M. F. Tasgetiren, Y. C. Liang, M. Sevkli, and G. A. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1930–1947, 2007.

[14] J. Salerno, "Using the particle swarm optimization technique to train a recurrent neural model," in *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence*, pp. 45–49, Newport Beach, Calif, USA, November 1997.

[15] P. F. Ribeiro and W. Kyle Schlansker, *A Hybrid Particle Swarm and Neuronal Network Approach for Reactive*, IEEE, Power Control, 2006.

[16] C. R. Eberthart and H. Xiaohui, *Human Tremor Analysus Using Particle Swarm Optimization*, Purdue School of Engineering and Technology, Indiana University Purdue University Indianapolis, Indianapolis, 1999.

[17] E. E. Omizegba and G. E. Adebayo, "Optimizing fuzzy membership functions using particle swarm algorithm," in *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, SMC 2009*, pp. 3866–3870, usa, October 2009.

[18] D. de la O, O. Castillo, and A. Meléndez, "Optimization of Fuzzy Control Systems for Mobile Robots Based on PSO," in *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, vol. 547 of *Studies in Computational Intelligence*, pp. 191–208, Springer International Publishing, Cham, 2014.

[19] R. Martínez-Soto, O. Castillo, L. T. Aguilar, and A. Rodríguez Díaz, "A hybrid optimization method with PSO and GA to automatically design Type-1 and Type-2 fuzzy logic controllers," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 2, pp. 175–196, 2015.

[20] L. Amador-Angulo and O. Castillo, "Comparison of the optimal design of fuzzy controllers for the water tank using ant colony optimization," *Studies in Computational Intelligence*, vol. 547, pp. 255–273, 2014.

[21] R. Fierro and O. Castillo, "Design of fuzzy control systems with different PSO variants," *Studies in Computational Intelligence*, vol. 451, pp. 81–88, 2013.

[22] R. Goering, "Matlab edges closer to electronic design automation world," *EE Times*.

[23] G. K. Venayagamoorthy and S. Doctor, "Navigation of mobile sensors using PSO and embedded PSO in a fuzzy logic controller," in *Proceedings of the Conference Record of the 2004 IEEE Industry Applications Conference; 39th IAS Annual Meeting*, pp. 1200–1206, usa, October 2004.

[24] J. Van Ast, R. Babuška, and B. De Schutter, "Fuzzy ant colony optimization for optimal control," in *Proceedings of the American Control Conference (ACC '09)*, pp. 1003–1008, IEEE, St. Louis, Mo, USA, June 2009.

[25] L. A. Zadeh, "Fuzzy logic," *The Computer Journal*, vol. 21, no. 4, pp. 83–93, 1988.

[26] H. Yoshida, Y. Fukuyama, S. Takayama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control in electric power systems considering voltage security assessment," in *Proceedings of the IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 497–502, Tokyo, Japan.

[27] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, vol. 1447 of *Lecture Notes in Computer Science*, pp. 591–600, Springer, New York, NY, USA, 1998.

[28] F. Valdez, P. Melin, and O. Castillo, "Particle swarm optimization for designing an optimal fuzzy logic controller of a DC motor," in *Proceedings of the 2012 Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS 2012*, usa, August 2012.

[29] T. Y. Abdalla, A. A. Abed, and A. A. Ahmed, "Mobile robot navigation using PSO-optimized fuzzy artificial potential field with fuzzy control," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 32, no. 6, pp. 3893–3908, 2017.

[30] H. A. Hashim, S. El-Ferik, and M. A. Abido, "A fuzzy logic feedback filter design tuned with PSO for adaptive controller," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9077–9085, 2015.

[31] M. Clerc, "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 3, pp. 1951–1957, 1999.

[32] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation and IEEE World Congress on Computational Intelligence*, (Cat. No.98TH8360), pp. 69–73, Anchorage, Alaska, USA, May 1998.

[33] J. Peng, Y. Chen, and R. Eberhart, "Battery pack state of charge estimator design using computational intelligence approaches," in *Proceedings of the 15th Annual Battery Conference on Applications and Advances*, pp. 173–177, usa, January 2000.

[34] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "Practical distribution state estimation using hybrid particle swarm optimization," in *Proceedings of the 2001 IEEE Power Engineering Society Winter Meeting*, pp. 815–820, usa, February 2001.

[35] Y. H. Shi and R. C. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, pp. 101–106, IEEE Press, May 2001.

[36] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, vol. 1 of *(Cat. No.00TH8512)*, pp. 84–88, July 2000.

[37] S.-M. Zhang, D. Huang, S.-C. Chu, T.-W. Sung, and T.-Y. Wu, "An adaptive ACO-based node deployment algorithm in

wireless sensor networks," *Journal of Internet Technology*, vol. 18, no. 5, pp. 1193–1202, 2017.

[38] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," Technical report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1991.

[39] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank based version of the ant system: a computational study," *Central European Journal of Operations Research*, vol. 7, no. 1, pp. 25–38, 1999.

[40] T. Stutzle and H. H. Hoos, "Improving the Ant System: A detailed report on the MAX-MIN Ant System," Technical report AIDA-96-12, FG Intellektik, FB Informatik, TU Darmstadt, Germany, 1996.

[41] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[42] O. Castillo, H. Neyoy, J. Soria, P. Melin, and F. Valdez, "A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot," *Applied Soft Computing*, vol. 28, pp. 150–159, 2015.

[43] O. Castillo and J. Soria, "Fuzzy Logic for Dynamic Parameter Tuning in ACO and Its Application in Optimal Fuzzy Logic Controller Desig," *Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics*, 2015.

[44] E. Vinodh Kumar, G. S. Raaja, and J. Jerome, "Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter," *Applied Soft Computing*, vol. 41, pp. 77–90, 2016.

[45] F. Olivas, F. Valdez, O. Castillo, C. I. Gonzalez, G. Martinez, and P. Melin, "Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems," *Applied Soft Computing*, vol. 53, pp. 74–87, 2017.