



# Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation

Camilo Caraveo, Fevrier Valdez, Oscar Castillo\*

Tijuana Institute of Technology, Tijuana Mexico

## ARTICLE INFO

### Article history:

Received 5 August 2015  
 Received in revised form  
 15 December 2015  
 Accepted 20 February 2016  
 Available online 27 February 2016

### Keywords:

Fuzzy logic  
 Fuzzy controller  
 Linguistic variables  
 BCO  
 FBCO  
 Dynamic parameter adjustment

## ABSTRACT

In this paper we are presenting a modification of a bio-inspired algorithm based on the bee behavior (BCO, bee colony optimization) for optimizing fuzzy controllers. BCO is a metaheuristic technique inspired by the behavior presented by bees in nature, which can be used for solving optimization problems. First, the traditional BCO is tested with the optimization of fuzzy controllers. Second, a modification of the original method is presented by including fuzzy logic to dynamically change the main parameter values of the algorithm during execution. Third, the proposed modification of the BCO algorithm with the fuzzy approach is used to optimize benchmark control problems. The comparison of results show that the proposed fuzzy BCO method outperforms the traditional BCO in the optimal design of fuzzy controllers.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

In optimization problems, the main objective is to find the best alternative among a set of possible solutions. In some cases the space of solutions for the problems is just too big and this can cause that the time to find the very best solution is prohibitive large. On the other hand, there are different areas of computational intelligence that provide a set of techniques for solving search and optimization problems [6,29,36]. Such techniques can provide highly competitive results, but not the best solutions. In addition there are alternative methods using heuristic algorithms, but these do not guarantee to find the best solution, although they are able to find a good solution in a reasonable time.

Population based algorithms constitute a new paradigm of collective intelligence and they are able to find good solutions to optimization problems with a reasonable cost and time [10,13,36]. These algorithms have become a research topic of recent interest to many scientists in the field of artificial intelligence [2,25,33]. Collective intelligence can be defined as a set of metaheuristic techniques of artificial intelligence based on the study of collective behavior systems present in nature, generally in a decentralized and self-organizing fashion [2].

There are many works on search and optimization algorithms that have been applied to a plethora problems, for example; in [1] an Artificial Bee Colony (ABC) algorithm to tune optimal rule-base of a Fuzzy Power System Stabilizer (FPSS) is presented, which leads to damp low frequency oscillation following disturbances in power systems. In [3] the optimization of the type-1 and type-2 fuzzy controller design for the water tank using the bee colony optimization method is presented. In [11] the design of an optimal fuzzy logic-PID controller using bee colony optimization for frequency control in an isolated wind-diesel system is presented. In [12] a bee colony optimization based-fuzzy logic-pid control

design of electrolyzer for microgrid stabilization is presented. In [39] an optimal fuzzy load frequency controller with simultaneous auto-tuned membership functions and fuzzy control rules is presented. In [8] the optimization of fuzzy controllers design using the bee colony algorithm is presented and the BCO algorithm is used to optimize the parameters of membership functions of a fuzzy logic controller.

The main contribution of this paper is the modification of the bee colony optimization algorithm (BCO), by adding the fuzzy approach to dynamically change its parameters. In the literature there are similar works using the bee colony algorithm to solve different problems, for example [1,11], where the main difference with respect to our proposal is the algorithm that was used. We use the BCO bee colony optimization (created by Teodorovic in 2002) [30–32] unlike to the algorithm used in [39], which is the artificial bee colony (created by Karaboga [19]), both bio-inspired algorithms are based on the behavior of bees, but with different features that are described below. The ABC algorithm has: follower bees, scout bees, and working bees, and the selection metric is different. The BCO algorithm has: follower bees, scout bees, and the process in the algorithm is different. In this case, the bees use a step forward and a step back, in step forward the bees explores new solutions, and in the step back the bees share information of new solutions with the bees of the hive, and this defines the role of follower bees or recruiter bees, therefore these are different algorithms.

There exists a previous publication by the authors using the same BCO algorithm with the difference that in the present paper, fuzzy logic for dynamic adjustment of parameters is now applied to different benchmark control problems. In the previous work, the algorithm is only used to adjust the parameters of the membership functions of a simple fuzzy controller.

The paper is organized as follows. The methodology is explained in Section 3, the theory and operation of the algorithm are described in Section 2, the cases used in this work are described in Section 4, the experimental results with the traditional method and with dynamic parameter adaptation using fuzzy logic and the statistical comparison between the two algorithms are described in Section 5. Finally, in Section 6 the conclusions are presented.

\* Corresponding author. Tel.: +52 6646236318.

E-mail address: [ocastillo@tectijuana.mx](mailto:ocastillo@tectijuana.mx) (O. Castillo).

## 2. Bee colony optimization (BCO)

The proposed algorithm is based on the artificial bee colony (BCO, bee colony optimization), which is one of the recently proposed algorithms in the area of collective intelligence [30,33,34]. The present paper focuses on the model proposed by Lucic and Teodorović in 2001, motivated by observing the intelligent behavior of honey bee's swarms [19,20,31,32].

Bee colony optimization (BCO) is a meta-heuristic algorithm [13,20] that belongs to the class of nature-inspired algorithms. These algorithms are inspired by various biological and natural processes [20]. Natural systems have become an important source of ideas and models for the development of many artificial systems [6].

This method (BCO) uses an analogy based on the way the bees do their foraging in nature, and the way in which they apply their search optimization methods to find optimal routes between the hive and the source food [20]. The basic idea behind BCO is to build the multi-agent system (artificial bee colony) [21] efficiently able to solve difficult combinatorial optimization problems [20,23].

### 2.1. Elements and behavior of BCO

The model defines three main components as shown below [20,21]:

- 1) Source of food: the value of a food source depends on many factors, including its proximity to the hive, wealth or food concentration and ease of food extraction.
- 2) Working bees: they are associated with a current or exploited food source. They carry with them information about that particular source, its distance, location and return to share, with a certain probability.
- 3) Scout bees: they are in constant search of a food source.

#### 2.1.1. Construction of a path by an artificial bee

In this model, a bee is allowed to explore and find a full path travel when leaving the hive the bees observe random dances performed by other bees. Then these bees are equipped with an array of movements of the observed dances.

This set of moves, referred to as "preferred path" is denoted as  $\theta$ , and it will serve as a guide in the process of foraging.  $\theta$  contains a complete path that was previously explored by its partner who will lead the bee to the destination.

During foraging, a bee goes from a node to another node until the destination is reached. In the model bee a heuristic rule is used to help transition the bee in its decision making about what node to visit next.

This rule is composed of two factors: the arc fitness and the distance heuristic. The arc fitness is calculated by Eq. (1) for all possible paths that can be visited by a bee on a particular node at a particular moment.

A bee is aided by a transition rule in decision making of the next node to visit, as is shown in Eq. (1). The state transition probability,  $P_{ij,n}$ , gives the probability of moving from node  $i$  to node  $j$  after  $n$  transitions. This is formally defined in Eq. (1) [37]

$$p_{ij,n} = \frac{[\rho_{ij,n}]^\alpha \cdot \left[\frac{1}{d_{ij}}\right]^\beta}{\sum_{j \in A_{i,n}} [\rho_{ij,n}]^\alpha \cdot \left[\frac{1}{d_{ij}}\right]^\beta} \quad (1)$$

where  $\rho_{ij,n}$  is the arc fitness from node  $i$  to node  $j$  after  $n$  transitions and  $d_{ij}$  represents the distance between node  $i$  and node  $j$ . Note that the  $P_{ij,n}$  is inversely proportional to the node distance. In other words, the shorter the distance, the higher is the likelihood of that

node to be selected.  $\alpha$  is a binary variable that turns on or off the arc fitness influence in the model.  $\beta$  is used to control the significance level of the heuristic distance [37].

The main references in the literature do not provide values for the alpha and beta parameters, and we consider using for beta the range from 2 to 7, and for alpha between 0 and 1 because of its similarity with the ant colony algorithm. The alpha parameter has more relevance than beta in the equation, because if alpha is null, the bee decides to use the heuristic distance, makes an assessment of the nearby nodes and determines which nodes to visit depending on the value of the fitness.

The arc fitness,  $\rho_{ij,n}$ , is defined as in Eq. (2). Where  $|A_{i,n} \cap F_{i,n}|$  is 1 when there is a common instance in both  $A_{i,n}$  and  $F_{i,n}$ , or 0 otherwise.  $A_{i,n} - F_{i,n}$  denotes the difference between sets  $A_{i,n}$  and  $F_{i,n}$ . It contains all elements of  $A_{i,n}$  that are not present in  $F_{i,n}$ . When there is only one node left in  $A_{i,n}$ ,  $\rho_{ij,n}$  is set to 1 to indicate that the node is the only remaining choice.

This happens at the last transition before a bee revisits the start node in order to complete the tour [20]

$$\rho_{ij,n} = \begin{cases} \lambda, & j \in F_{i,n}, |A_{i,n}| > 1 \\ \frac{1 - \lambda|A_{i,n} \cap F_{i,n}|}{|A_{i,n} - F_{i,n}|}, & j \notin F_{i,n}, |A_{i,n}| > 1 \\ 1, & |A_{i,n}| = 1 \end{cases} \quad \forall j \in A_{i,n}, \quad (2) \\ \mathbf{0} \leq \lambda \leq 1$$

## 3. Algorithm with dynamic adjustment using Fuzzy Logic

This section is dedicated to describing the modification of the BCO algorithm with a fuzzy logic approach for dynamically adjusting the values of the  $\alpha$  and  $\beta$  parameters. The proposed methodology is described below [15,18,26,28,37]. The particular objective in choosing the BCO algorithm is because there are few research works published of this algorithm, there also exist other variants and we consider a good idea to use it and analyze if good results can be achieved when compared with respect to other algorithms.

### 3.1. Method for adaptation of parameters

In Fig. 1 we graphically illustrate the sequence of steps of the proposed algorithm:

The pseudocode of the proposed algorithm is presented as follows:

Step 1: Randomly generate the initial population of  $n$  scout bees for the MF parameters.

To start the algorithm a set of random source foods is generated, where each row (vector) represents a bee and the bee has in memory a route and that route is a possible solution to the problem, which in this case represents the values of the membership function parameters of the fuzzy controller.

This initial population must have possible candidate solutions that satisfy the constraints. Set  $NC=0$ , and evaluate the fitness value of the initial populations by (6).

Step 2: Select the  $m$  best sites for the neighborhood search.

The bee selected memory contains the best path found so far, and to define the fitness of each bee the equation of the mean square error is used, which is Eq. (9).

Step 3: Recruit bees are used for selected sites (more bees for the best sites).

Step 4: Represent the new value of MF from each working bee.

Step 5: Select the fittest bees from each path.

Step 6: Check the stopping criterion. If satisfied, terminate the search, else  $NC = NC + 1$ , show and save the best values found and return to evaluate new solutions.

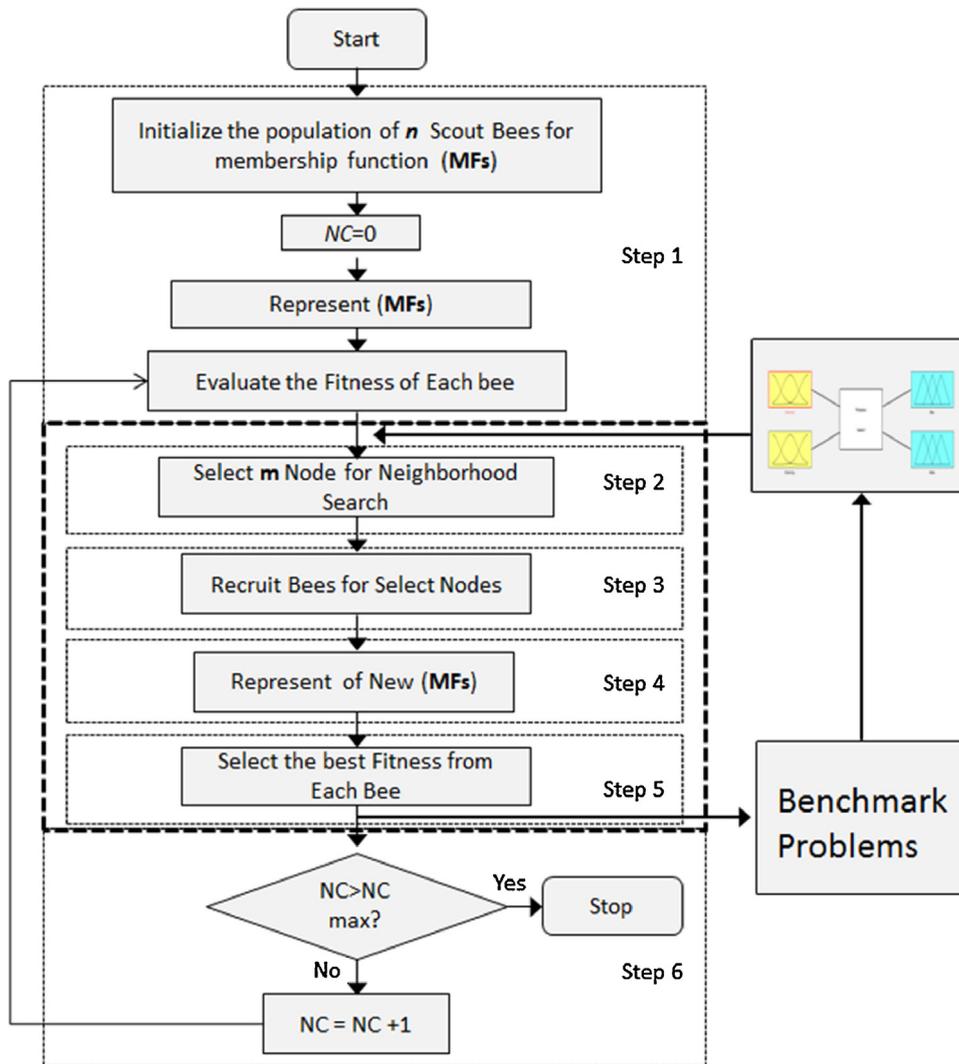


Fig. 1. Procedure for tuning parameters in the BCO algorithm.

The modification to the algorithm is made in Eq. (1). This function is used to calculate the transition probability,  $P_{ij,n}$ , and gives the probability of moving from node  $i$  to node  $j$  after  $N$  transitions [37].

The parameters that are dynamically adjusted with the help of fuzzy logic are  $\alpha$  and  $\beta$ , where:  $\alpha$  is a parameter that turns on or off the arc fitness influence in the model and  $\beta$  is used to control the significance level of the heuristic distance.

We proposed to use the following fuzzy control architecture for the dynamic variation of parameters at runtime, as is illustrated in Fig. 2.

The fuzzy logic controller used for dynamically adjusting the algorithm parameters is of Mamdani type with 2 inputs, 2 outputs and 9 rules that are chosen with prior knowledge of the problem of exploration and exploitation in search spaces.

In this case we decided to use as an input the iteration variable of the algorithm to measure the state of the iterations. In this case when the algorithm starts we consider the iterations to be “low” and when the iterations are completed, then are considered “high” or near 100%. To represent this idea we use the following Eq. (4) [26], the input variable considers the number of

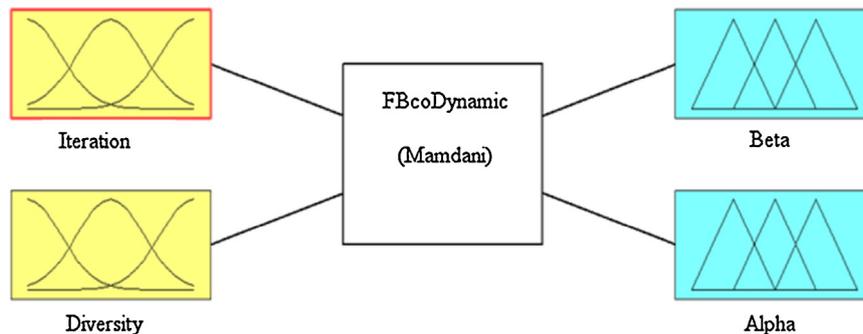


Fig. 2. Structure of the fuzzy system for dynamic adjustment of the parameters.

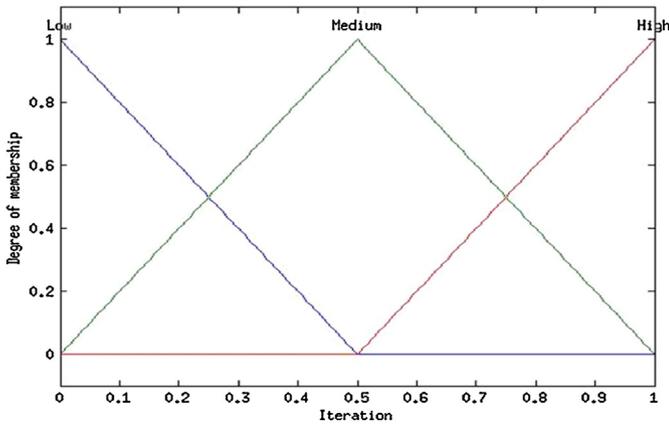


Fig. 3. Iteration input variable.

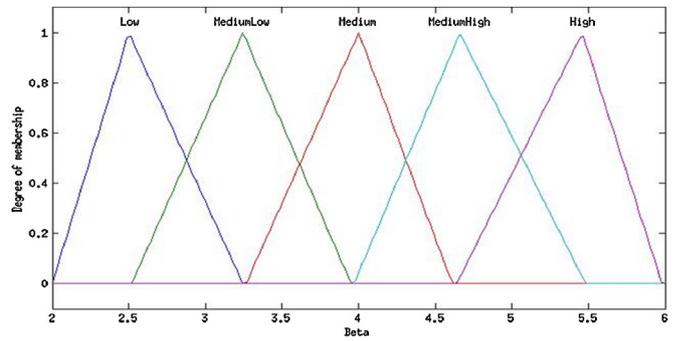


Fig. 5. Beta output variable.

iterations of the algorithm. The granulation of the variable is shown in Fig. 3.

The second input variable is the diversity defined by Eq. (5), which measures the degree of dispersion of the bees, when the bees are closer together, there is less diversity, and when bees are separated there is high diversity. The diversity equation can be viewed as the average of the Euclidean distances between each bee and the best bee, the diversity input variable takes the value of dispersion of the bees and the values are changing in each iteration of the algorithm. The granulation of this variable is shown in Fig. 4, the range of values is from 0 to 100 and are defined using Eq. (4), which shows the membership functions of the diversity input variable [26]. In this case  $x_{ij}$  represents the current solution and,  $z_j$  represents the best global solution.

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

$$\text{Diversity}(S(t)) = \frac{1}{n_s} \sum_{i=1}^n \sqrt{\sum_{j=1}^{n_x} (x_{ij}(t) - z_j(t))^2} \quad (5)$$

For the  $\alpha$  and  $\beta$  output variables, as mentioned above, the recommended values for  $\alpha$  (0–1) and  $\beta$  (2–7), so that the output variables are granulated using these ranges [8]. The beta output is granulated into five triangular membership functions, see Fig. 5, and the second output variable Alpha is granulated into four Gaussian membership functions, see Fig. 6. Having defined the possible input and output variables, then it was decided to combine them to generate the different output values of  $\alpha$  and  $\beta$ .

We show below the set of fuzzy rules defined in the proposed approach:

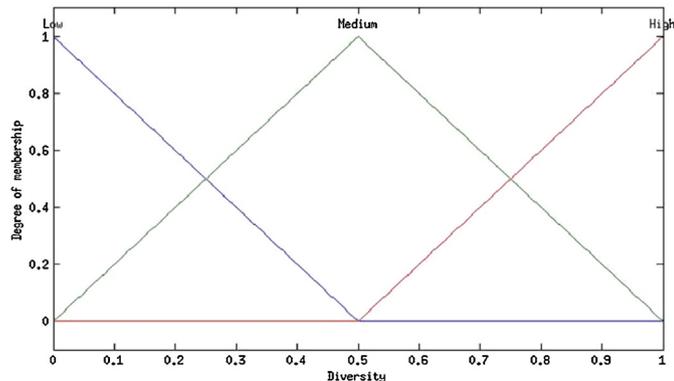


Fig. 4. Diversity input variable.

1. If (Iteration is Low) and (Diversity is Low) then (Beta is Medium)(Alpha is Regular)
2. If (Iteration is Low) and (Diversity is Medium) then (Beta is High)(Alpha is High)
3. If (Iteration is Low) and (Diversity is High) then (Beta is MediumLow)(Alpha is Low)
4. If (Iteration is Medium) and (Diversity is High) then (Beta is MediumHigh)(Alpha is Regular)
5. If (Iteration is Medium) and (Diversity is Medium) then (Beta is Medium)(Alpha is High)
6. If (Iteration is Medium) and (Diversity is Low) then (Beta is MediumHigh)(Alpha is Low)
7. If (Iteration is High) and (Diversity is Low) then (Beta is Low)(Alpha is High)
8. If (Iteration is High) and (Diversity is Medium) then (Beta is High)(Alpha is Regular)
9. If (Iteration is High) and (Diversity is High) then (Beta is MediumHigh)(Alpha is Regular)

For this problem a set of 9 rules was used to define the adaptation of parameters, the rules are designed with the help of previous knowledge based on performed experiments, where it was observed that when this combination of rules is used improved results are obtained, so we decided to use this combination of fuzzy rules. The main issue is achieving the correct combination of rules for obtaining the appropriate control of the parameter values in BCO.

#### 4. Benchmark problems

It is very important for the reader to know the problems that are considered in this work for their easy understanding, and this section provides a brief description of the different benchmark control problems that are used to test the performance of the proposed

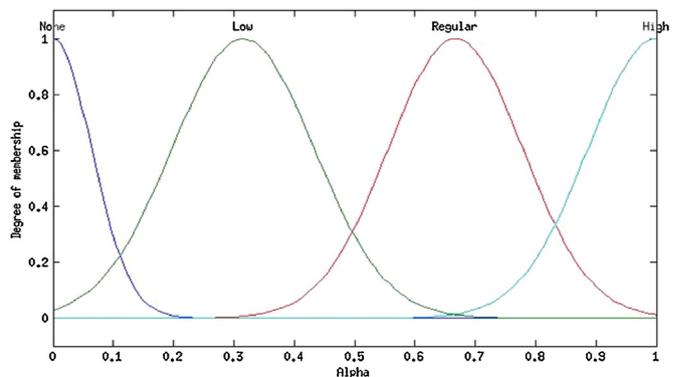


Fig. 6. Alpha output variable.

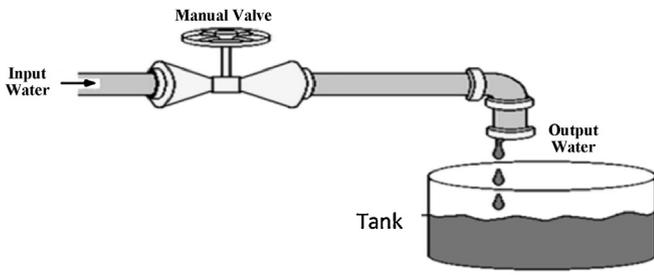


Fig. 7. Graphic representation of the problem.

algorithm. To test the performance of the controllers a mathematical model of the plant is used in each case [5,35], and in this form we can evaluate if the corresponding fuzzy controller is fulfilling its function.

4.1. Fuzzy controller for the benchmark water tank problem

The first problem is known as the water tank controller, which can be described as follows: the controller aims at a particular water level in a tank, therefore, we have to know the actual water level in the tank and with that we have to be able to set the valve. Fig. 7 shows graphically the way in which the valve opening operates and hence the filling process in the tank. This will have two variables which are the water level as input and the speed of opening of the valve tank filling is the output [4,5,8].

The main goal in this case is to design a fuzzy controller for maintaining the water level in a tank.

4.1.1. Characteristics of the fuzzy controller

We now present the characteristics of the fuzzy controller that was used, which is based on [35]. The controller has two inputs to the fuzzy system [38]: the first is the water level, which has three membership functions with linguistic values of “high”, “okay” and “low”, as shown in Fig. 8. The second input variable is called rate with three membership functions with linguistic values of “negative”, “good” and “positive”, and these are shown in Fig. 8. The names of the linguistic labels are assigned based on the empirical process of the filling behavior of a water tank [5,8,18].

The fuzzy inference system has an output variable called “valve”, which has five “triangular membership” functions with the following linguistic values: “close\_fast”, “close\_slow”, “no\_change”, “open\_slow” and “open\_fast”.

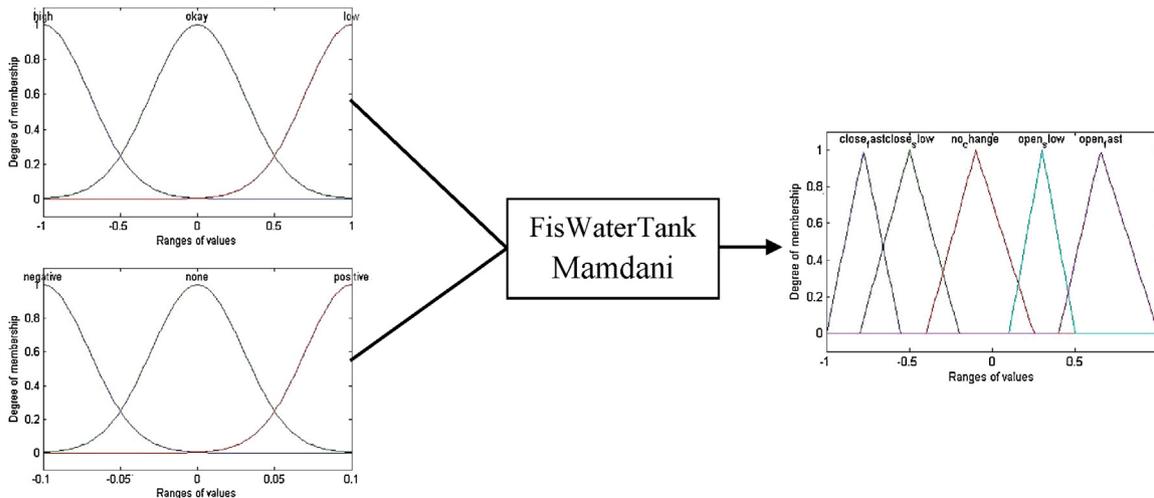


Fig. 8. General structure of the fuzzy system with the input and output variables.

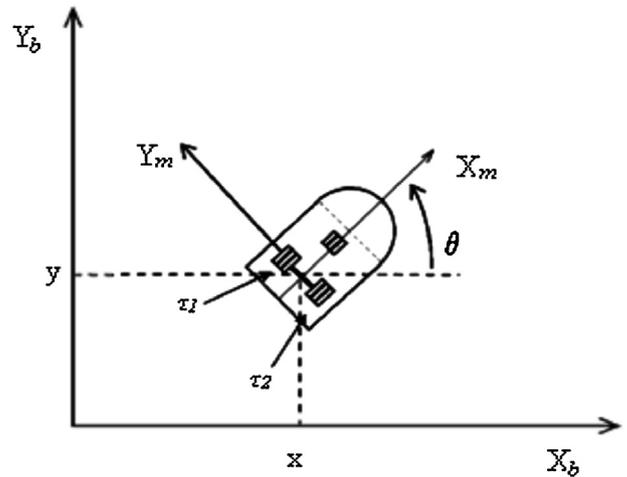


Fig. 9. Representation of the mobile robot.

The fuzzy rules of the controller are:

- If (level is okay) then (valve is no\_change).
- If (level is low) then (valve is open\_fast).
- If (level is high) then (valve is close\_fast).
- If (level is okay) and (rate is positive) then (valve is close\_slow).
- If (level is okay) and (rate is negative) then (valve is open\_slow).

The combination of rules was found experimentally according to the theory of how the process for filling a water tank is performed. We consider these 5 rules to display the behavior of the fuzzy controller [18,27].

4.2. Autonomous mobile robot

The mobile robot that was used as a case study, is an autonomous vehicle capable of following predictable paths in uncertain environments and is illustrated in Fig. 9.

The robot body is symmetrical around the perpendicular axis and the center of mass is at the geometric center of the body. It has two driving wheels that are fixed to the axis that passes through the center of mass “C” represented by {C, X<sub>m</sub>, Y<sub>m</sub>}, and one passive wheel that prevents the robot from tipping over as it moves on a

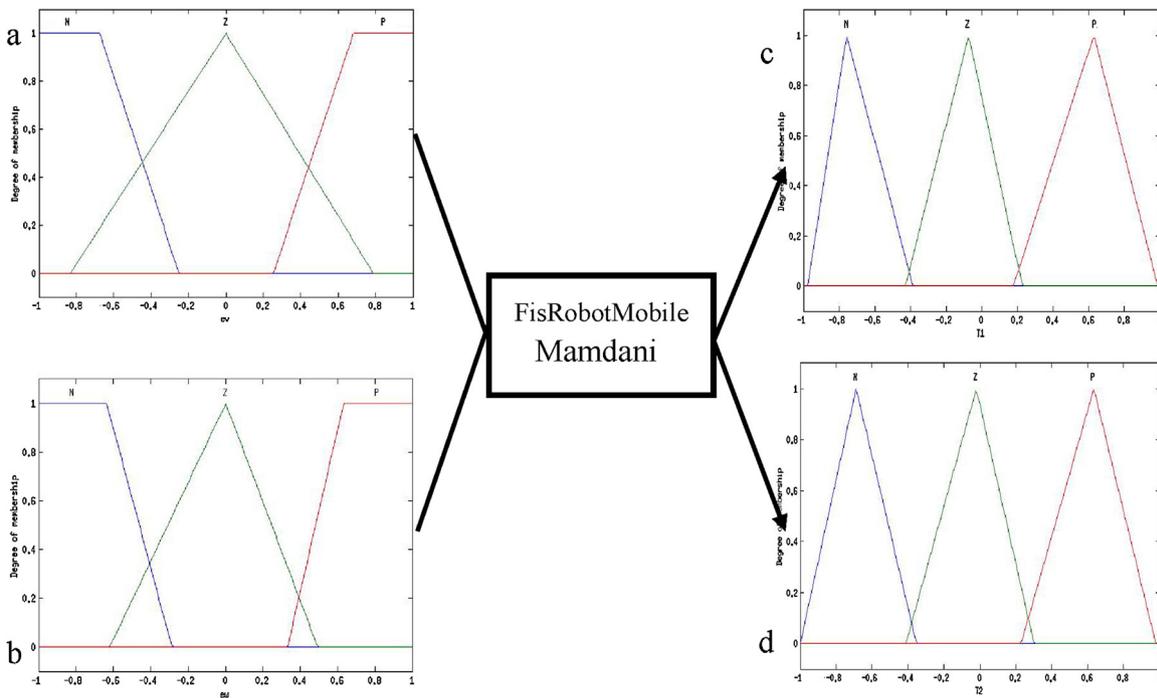


Fig. 10. (a) Linear velocity error ( $e_v$ ), (b) angular velocity error ( $e_w$ ), (c) left torque ( $T_1$ ), (d) right torque ( $T_2$ ).

plane [9,10]. The dynamics of the mobile robot is represented by Eqs. (6) and (7) [9,10].

$$M(q)\dot{v} + C(q, \dot{q})v + Dv = \tau + P(t) \quad (6)$$

where,

$q = (x, y, \theta)^T$  is the vector of the configuration coordinates,  $v = (v, w)^T$  is the vector of velocities,  $\tau = (\tau_1, \tau_2)$  is the vector of torques applied to the wheels of the robot where  $\tau_1$  and  $\tau_2$  denote the torques of the right and left wheel,  $P \in R^2$  is the uniformly bounded disturbance vector,  $M(q) \in R^{2 \times 2}$  is the positive-definite inertia matrix,  $C(q, \dot{q})v$  is the vector of centripetal and Coriolis forces, and  $D \in R^{2 \times 2}$  is a diagonal positive-definite damping matrix.

The kinematic system is represented by Eq. (7)

$$\dot{q} = \underbrace{\begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_v \quad (7)$$

where,

$(x, y)$  is the position in the X–Y (world) reference frame,  $\theta$  is the angle between the heading direction and the x-axis,  $v$  and  $w$  are the linear and angular velocities.

Furthermore Eq. (8) shows the non-holonomic constraint, which this system has, which corresponds to a non-slip wheel condition preventing the robot from moving sideways.

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (8)$$

The system fails to meet Brockett's necessary condition for feedback stabilization, which implies that no continuous static state-feedback controller exists that can stabilize the closed-loop system around the equilibrium point.

#### 4.2.1. Characteristics of the fuzzy controller used for the robot

Below the characteristics of the fuzzy controller used in this case study are presented. We chose to optimize a fuzzy controller of the path for a mobile unicycle robot to test the proposed method in a more complex problem. The controller is of Mamdani type, so that the input and output parameters are represented by linguistic variables. The input variables are the error in the linear velocity ( $e_v$ ) and angular velocity ( $e_w$ ), and the output variables are the right ( $T_1$ ) and left ( $T_2$ ) torques [9,10], which are represented in Fig. 10.

The membership functions of the input variables are of trapezoidal type in the negative (N) and positive (P) linguistic terms and triangular for the linguistic term for zero (Z). For the output variables we have 3 membership functions, negative (N), zero (Z), positive (P) of triangular type. The range  $[-1, 1]$  is used for each variable because we have normalized all the variables.

The 9 fuzzy rules that are used are shown below in Fig. 11:

#### 4.3. Temperature control problem

In this section the benchmark problem of temperature control, that is also used to test the proposed method, is described in

1. If ( $e_v$  is N) and ( $e_w$  is N) then ( $T_1$  is N)( $T_2$  is N)
2. If ( $e_v$  is N) and ( $e_w$  is Z) then ( $T_1$  is N)( $T_2$  is Z)
3. If ( $e_v$  is N) and ( $e_w$  is P) then ( $T_1$  is N)( $T_2$  is P)
4. If ( $e_v$  is Z) and ( $e_w$  is N) then ( $T_1$  is Z)( $T_2$  is N)
5. If ( $e_v$  is Z) and ( $e_w$  is Z) then ( $T_1$  is Z)( $T_2$  is Z)
6. If ( $e_v$  is Z) and ( $e_w$  is P) then ( $T_1$  is Z)( $T_2$  is P)
7. If ( $e_v$  is P) and ( $e_w$  is N) then ( $T_1$  is P)( $T_2$  is N)
8. If ( $e_v$  is P) and ( $e_w$  is Z) then ( $T_1$  is P)( $T_2$  is Z)
9. If ( $e_v$  is P) and ( $e_w$  is P) then ( $T_1$  is P)( $T_2$  is P)

Fig. 11. Combination of fuzzy rules used for control.

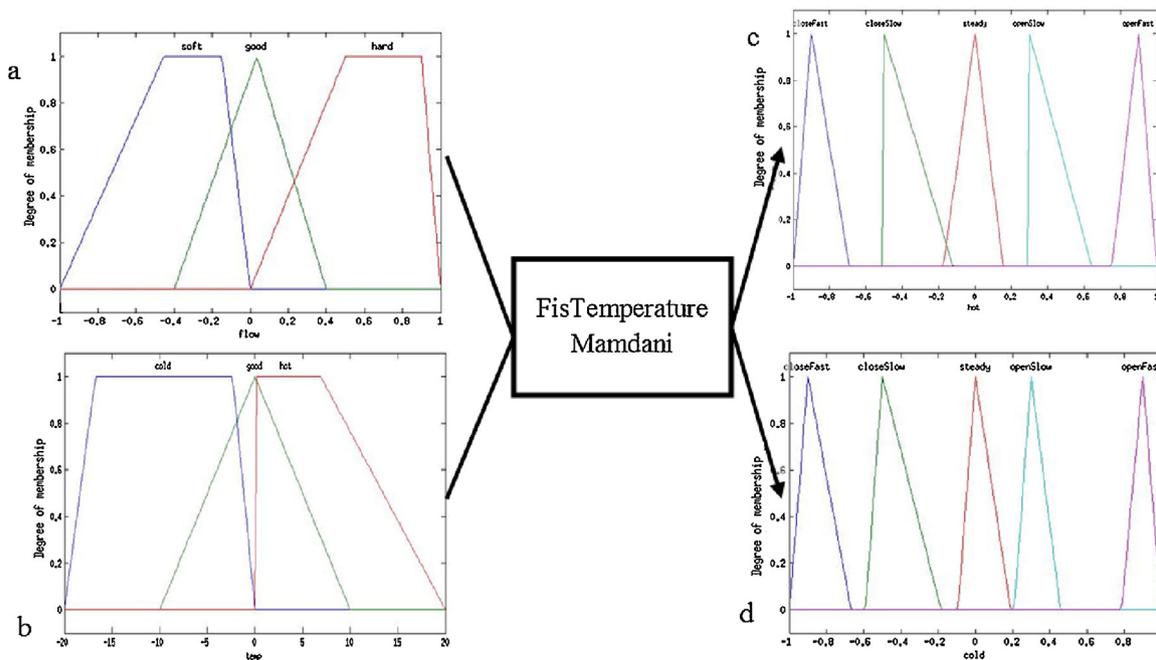


Fig. 12. (a) Flow, (b) temp, (c) hot, (d) cold.

detailed fashion. The main problem in this case study is to control the temperature in the water flow [17].

4.3.1. Characteristics of the fuzzy controller used

Below the characteristics of the fuzzy controller used in this work are presented. The controller is of Mamdani type, and the input and output parameters are represented by linguistic variables. The linguistic values of the two inputs and output variables, are called “temp”, “flow” for the inputs and “cold” and “hot” for the outputs, which are shown in Fig. 12.

The membership functions for the inputs are of trapezoidal and triangular type and the output variables are of triangular form, as illustrated in Fig. 12. We choose to use the 9 fuzzy rules that are usually considered in the control of this plant, and the combination of rules are shown in Fig. 13.

4.4. Inverted pendulum control problem

The inverted pendulum control problem is a standard test application for complex control approaches, with its complexity originating from the nonlinear nature of the problem [14,22]. The cart pole balancing problem is a popular demonstration of using feedback control to stabilize an open-loop unstable system with fewer control inputs than the degrees of freedom. The cart-pole task involves a balancing pole hinged to a motionless cart that travels left or right along a straight bounded track as shown in Fig. 14. The

pole is free to rotate only in the vertical plane of the cart and track. There are no sidelong resultant forces on the pole and it remains balanced as shown in Fig. 14 [22].

The control objective is to apply a sequence of left or right forces of fixed magnitude to the wheeled cart so that it swings up the pendulum from its natural pendant position and stabilizes it to the inverted position; once it reaches the upright equilibrium point [14].

4.4.1. Characteristics of the fuzzy controller used for this problem

Below the characteristics of the fuzzy controller used in this work are presented. The controller can be of Takagi-Sugeno type. In this case study we will be using controllers of Sugeno type and measure the performance to find out if the approach is good for this problem. The general scheme is illustrated in Fig. 15 [22].

The controller used in this case study is of Sugeno type, and we consider as inputs four variables with membership functions of generalized bell type and the output is defined by mathematical functions. The function controller output represents the force that should push the car to maintain balance in the pendulum [7].

1. If (temp is cold) and (flow is soft) then (cold is openSlow)(hot is openFast)
2. If (temp is cold) and (flow is good) then (cold is closeSlow)(hot is openSlow)
3. If (temp is cold) and (flow is hard) then (cold is closeFast)(hot is closeSlow)
4. If (temp is good) and (flow is soft) then (cold is openSlow)(hot is openSlow)
5. If (temp is good) and (flow is good) then (cold is steady)(hot is steady)
6. If (temp is good) and (flow is hard) then (cold is closeSlow)(hot is closeSlow)
7. If (temp is hot) and (flow is soft) then (cold is openFast)(hot is openSlow)
8. If (temp is hot) and (flow is good) then (cold is openSlow)(hot is closeSlow)
9. If (temp is hot) and (flow is hard) then (cold is closeSlow)(hot is closeFast)

Fig. 13. Combination of rules used in the controller.

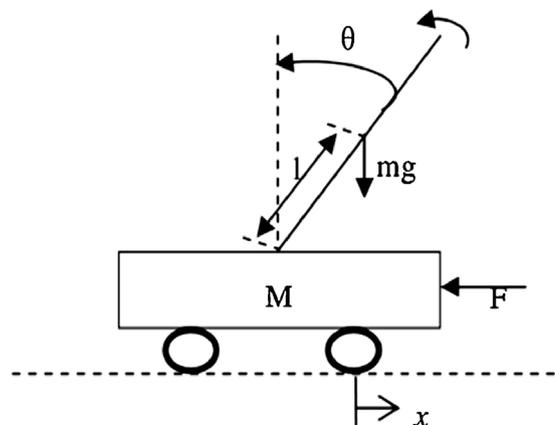


Fig. 14. Inverted pendulum system.

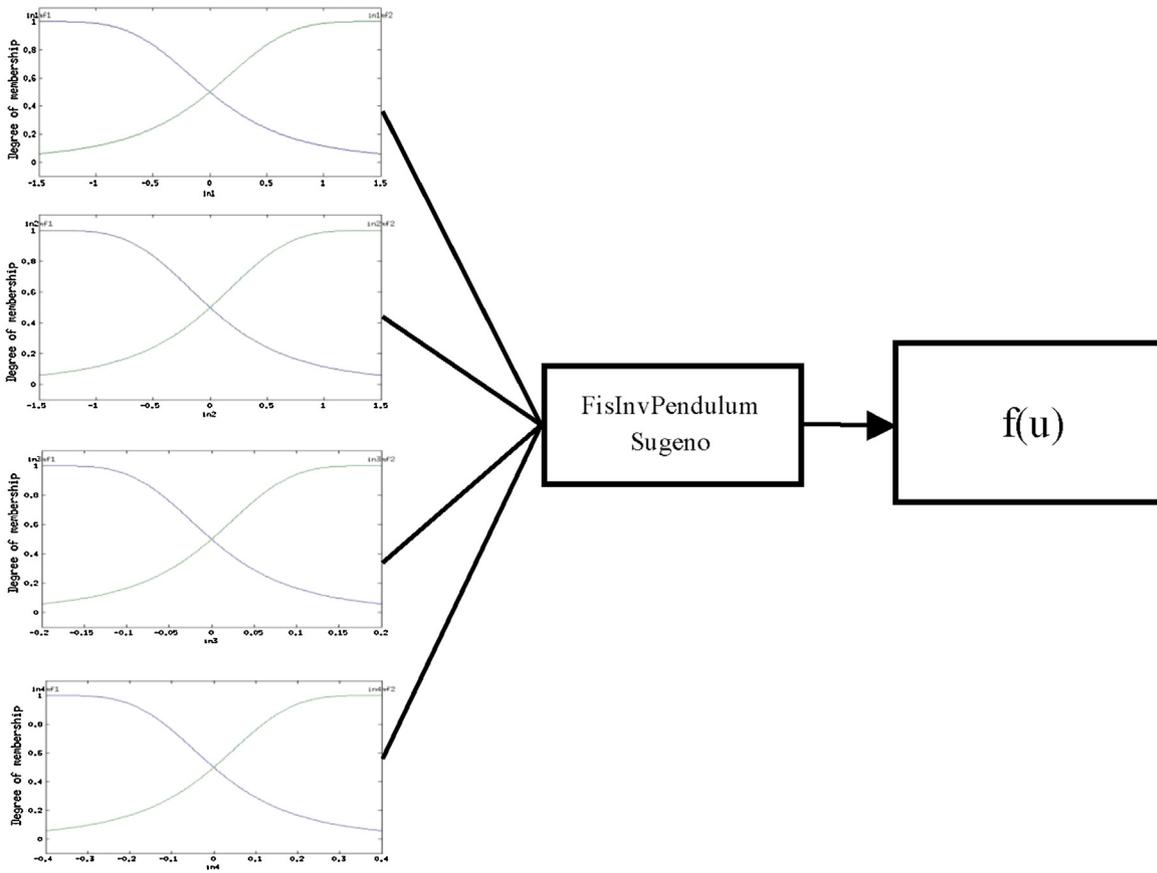


Fig. 15. General structure of the fuzzy controller.

5. Simulation results

This section presents the results obtained by optimizing the parameters of the membership functions of the input and output variables for the fuzzy controllers and the experiments were performed by dynamically changing the  $\alpha$  and  $\beta$  parameters.

The parameters recommended in the literature [24] for  $\alpha=(0-1)$ . In the literature, there are no recommended values for  $\beta$ , and we can find that the optimal values needed to achieve good results in the experiments are shown in the publication of Wong [24]. We used a value of  $\beta = 10$ , as suggested in [16,28].

The presented values are results of 30 experiments varying the values of the  $\alpha$  and  $\beta$  parameters. In this case the equation for calculating the fitness value of each bee, see Eq. (2), is replaced by the equation of the mean squared error (MSE), see Eq. (9), so we measure the performance of the fuzzy controller [8] designed by the BCO algorithm bee colony with this function. The equation is as follows:

$$MSE = \frac{1}{N} \sum_{K=1}^N [x(k) - y(k)]^2 \tag{9}$$

In Eq. (9)  $x(k)$  represents the reference value at time  $k$ ,  $y(k)$  represents the value produced by the system at time  $k$ , and  $N$  represents total number of samples considered.

The proposed algorithm is applied to different optimization problems, which are the benchmark control problems mentioned in Section 4, and the performance of the algorithm is considered in its original form and with the proposed method with the fuzzy approach to observe if there are some significant differences and justify the advantages of the proposed method.

5.1. BCO and FBCO applied to the problem of temperature control

In this section the results obtained by applying the proposed approach to the problem of temperature control are presented. This problem is widely used in the area of control as a benchmark and for more details, see Section 4.3. In Table 1 the most important data of 30 experiments are presented, and we have to say that in optimization problems it is important to report the following information: *Number iterations,  $\alpha$ ,  $\beta$ ,  $\sigma$ , average time, best and average* the experiments.

The experiments are performed with the same configuration parameters, unlike the fuzzy approach where the  $\alpha$  and  $\beta$  parameters are changed dynamically at runtime. In this case study, our proposal exceeds the expectations of the traditional bee colonies algorithm. In Fig. 16 the best fuzzy controller architecture found by the algorithm is presented.

We can observe a better performance of the fuzzy controller built by the proposed method (FBCO), where the black line represents the reference and the gray line represents the performance of the controller built to follow the reference.

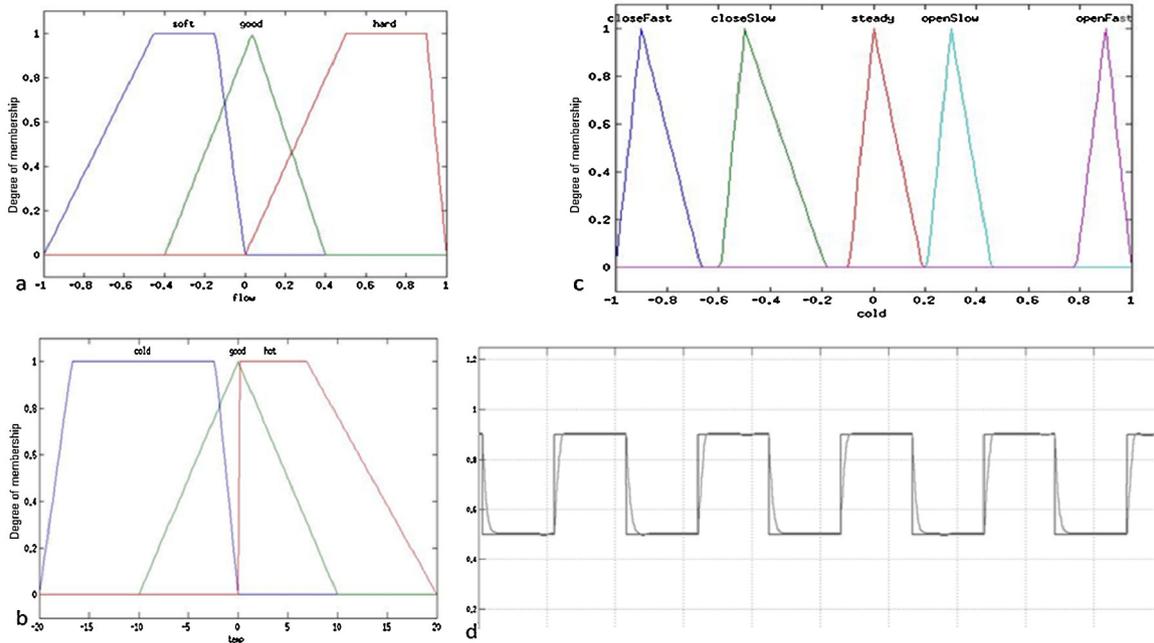
5.2. BCO and FBCO applied to the autonomous mobile robot

In this section the results of the experiments performed for the autonomous mobile robot problem are presented, and for more details on the robot see Section 4.2. In Table 2 the results of 30 experiments and the parameter configuration are presented.

In Fig. 17 a general diagram of the distribution of the membership functions of the best fuzzy controller for this case study is presented. In Fig. 17(e) we can observe a better performance of the fuzzy controller built by the proposed method (FBCO), where the blue line represents the reference and the green line represents the

**Table 1**  
Configuration parameters and results of BCO and FBCO (“Dyn” means dynamic).

Configuration parameters				Results			
Approach	Iterations	$\alpha$	$\beta$	$\sigma$	Average time	Best	Average
BCO	60	0.9	1	4.46E–04	6.72	3.40E–05	4.93E–04
FBCO	60	Dyn	Dyn	2.90E–04	7.59	1.62E–05	2.82E–04



**Fig. 16.** (a) Flow, (b) temp, (c) cold, (d) simulation.

**Table 2**  
Configuration parameters and results of BCO and FBCO.

Configuration parameters				Results			
Approach	Iterations	$\alpha$	$\beta$	$\sigma$	Average time	Best	Average
BCO	60	0.01	2	0.16811	8.7	4.61E–03	0.228
FBCO	60	Dyn	Dyn	0.06463	11.72	2.06E–03	0.068

trajectory of the controller built to follow the reference. For this problem, the proposed algorithm has greater stability and adaptation exceeding the performance of the traditional BCO method, and the results published by other authors demonstrated not to overcome our proposed approach [9,10].

**5.3. BCO and FBCO applied to water level control in a tank**

The algorithms (BCO and FBCO) were also applied to the problem of the water tank control. This problem is widely used as a benchmark to test control methods. In Table 3 the results and configuration parameters are presented.

In Fig. 18 a general diagram of the distribution of the membership functions of the best fuzzy controller for this case study is presented. In Fig. 18(e) we observe a better performance of the

fuzzy controller built by the proposed method (FBCO), where the blue line represents the reference and the green line represents the trajectory of the controller built to follow the reference. Many results by other authors with different methods have been published in [4,5,8], and in this case some have find better results than our proposal.

**5.4. BCO and FBCO applied to the inverted pendulum problem**

We also applied the algorithms in a different case, which is the problem of the inverted pendulum. In this case the fuzzy controller that is used is of Sugeno type, see Section 4.4 for more details. The idea is to test the performance of the algorithm for various problems of fuzzy control. In this section the results of the experiments are presented in Table 4.

**Table 3**  
Configuration parameters and results of BCO and FBCO.

Configuration parameters				Results			
Approach	Iterations	$\alpha$	$\beta$	$\sigma$	Average time	Best	Average
BCO	60	0.3	5	5.95E–03	11.25	6.16E–02	7.21E–02
FBCO	60	Dyn	Dyn	1.42E–02	10.45	1.03E–02	3.69E–02

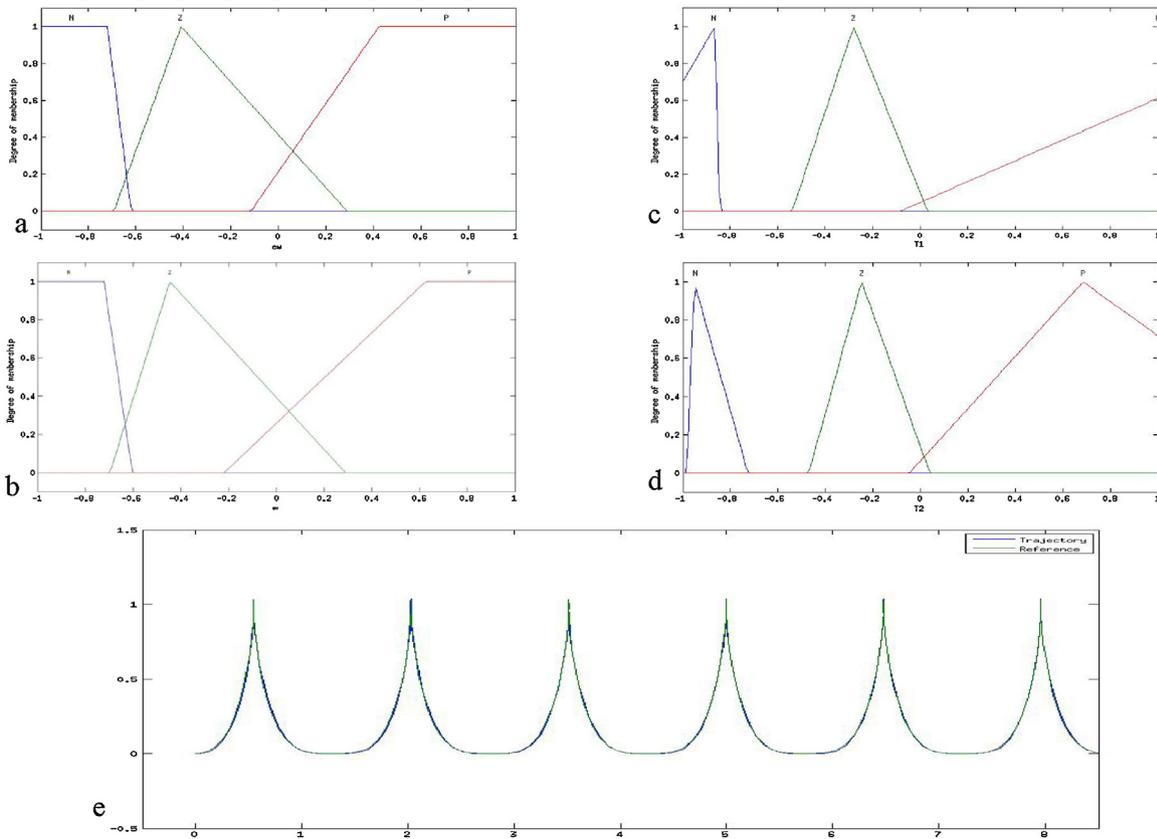


Fig. 17. (a)  $e_w$ , (b)  $e_v$ , (c) T1, (T2) T2, (e) simulation of the controller. (For interpretation of the references to colour in the citation of this figure, the reader is referred to the web version of this article.)

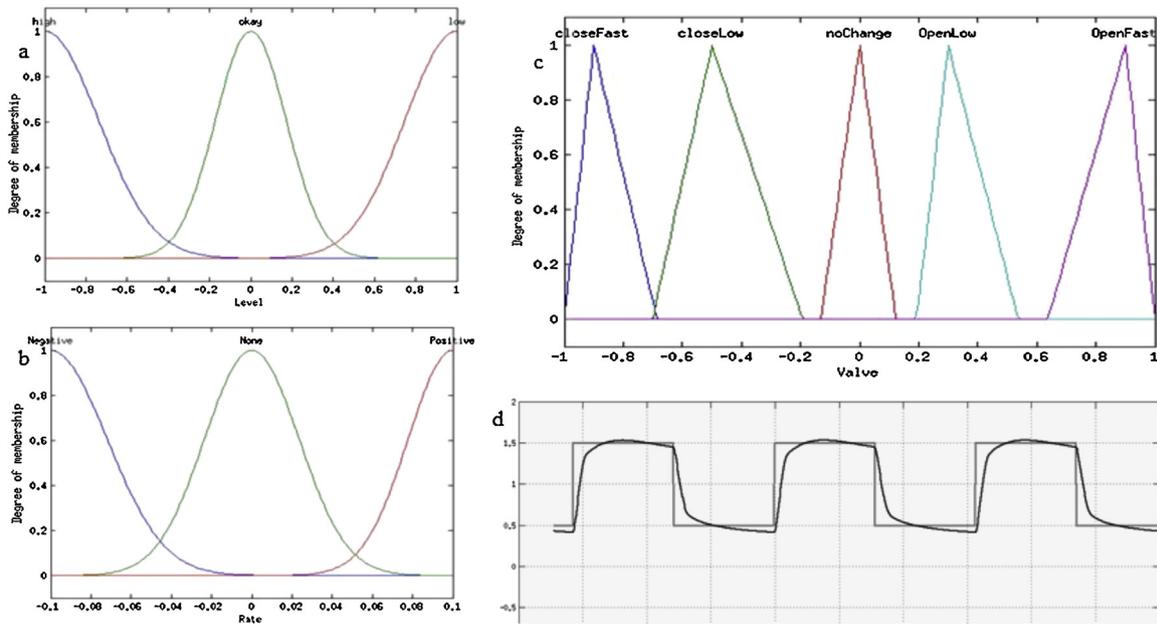


Fig. 18. (a) Level, (b) rate, (c) valve, (d) simulation results. (For interpretation of the references to colour in the citation of this figure, the reader is referred to the web version of this article.)

**Table 4**  
Configuration parameters and results of BCO and FBCO.

Configuration parameters				Results			
Approach	Iterations	$\alpha$	$\beta$	$\sigma$	Average time	Best	Average
BCO	60	0.1	3.0	0.11203	33.71	1.2217	1.366781
FBCO	60	Dyn	Dyn	0.07637	35.97	1.2120	1.432

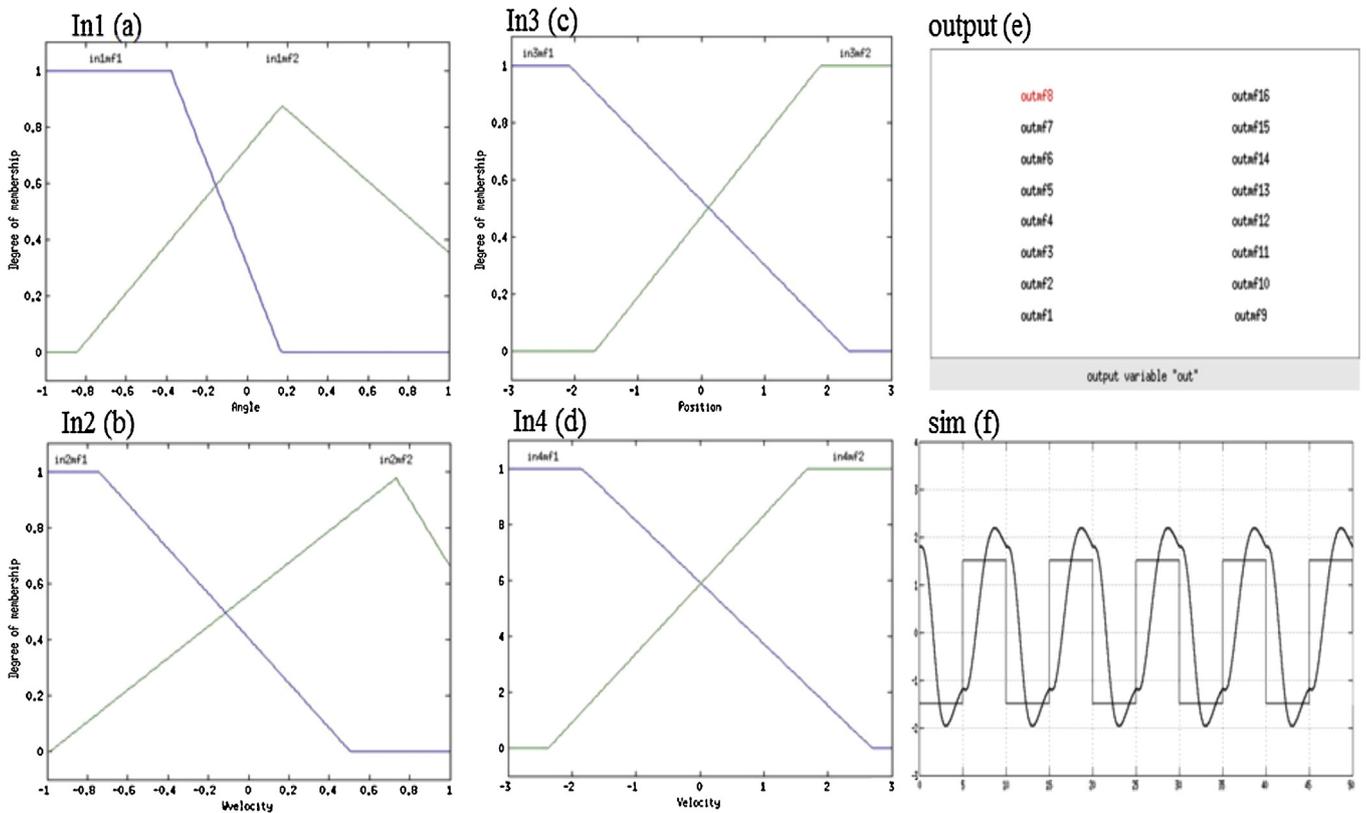


Fig. 19. (a) Angle, (b) wvelocity, (c) position, (d) velocity, (e) out, (f) simulation results

In Fig. 19 a general diagram of the architecture constructed by the algorithm is presented, and unlike the above mentioned controllers, in this case calculating the output values is made with mathematical functions. In the simulation results we can notice that the algorithm is not as good when compared to the other control problems. In Table 4 we can note that our proposal does not improve on the simple BCO algorithm for this case study. In fact, we have to say that both algorithms had difficulty to reach the target, and we can notice that there is no benefit in applying the proposed algorithm with the fuzzy approach for this type of fuzzy controllers and the computational cost is also high [14].

5.5. Statistical comparison

To conclude in a formal way, it is necessary to perform a statistical comparison between the traditional BCO vs FBCO for the different problems that were considered in this study. The statistical test that was used for the comparisons is the z-test, whose parameters are defined in Table 5. With the parameters of Table 5, we applied the statistical z-test for all the cases shown in this study giving the results shown in Table 6.

In applying the statistical z-test, with a significance level of 0.05, the alternative hypothesis states that the average of the proposed method is lower than the average of the traditional BCO, and of course the null hypothesis states that the average of the proposed

Table 5

Parameters for the statistical z-test.

Parameter	Values
Level of significance	95%
Alpha	0.05%
Ha	$\mu_1 < \mu_2$
H0	$\mu_1 \geq \mu_2$
Critical value	-1.645

method is greater than or equal to the average of the traditional BCO, with a rejection region for all values falling below -1.645. So the statistical test results are that: for the fuzzy BCO, there is significant evidence to reject the null hypothesis with a value of -2.1723.

Analyzing the results of the performed hypothesis tests shows that our proposed approach improves results with respect to the traditional method, with the exception to the problem of the inverted pendulum. The main reason for not improving results in the last case is that between the FBCO and BCO methods, we do not have sufficient statistical evidence to reject the null hypothesis, and this could be because the case study uses a Sugeno type fuzzy controller, which in many cases can be more difficult to design. We found that by using fuzzy logic in the bee colony algorithm the results are better when compared with the traditional method and we can conclude that with the addition of the dynamic parameter

Table 6

Results of applying the statistical z-test.

Case study	Our method	Simple method	Z-value	Evidence
Water tank problem	FBCO	BCO	-12.4933	Significant
Autonomous mobile robot	FBCO	BCO	-4.8806	Significant
Temperature control problem	FBCO	BCO	-2.1723	Significant
Inverted pendulum problem	FBCO	BCO	3.4913	Not significant

tuning we can maintain a better balance between the exploration and exploitation and with this avoiding premature convergence of the algorithm.

## 6. Conclusions

In this work we show the application of a collective intelligence paradigm, the bee colony algorithm (BCO) for solving optimization problems, in particular applied to benchmark control problems based on fuzzy logic. We conclude that dynamically adjusting parameters of an optimization method can improve the quality of results and can increase the diversity of solutions to a problem and the results can be improved when compared with already published results on the same control problems [8]. The main contribution was to now include a fuzzy approach to the bee colony algorithm using fuzzy logic to move the alpha and beta parameters dynamically at runtime. It is found that FBCO is an optimization method, which can be applied to any problem, and with the addition of dynamic parameter tuning we can maintain a better balance between the exploration and exploitation and with this avoiding premature convergence. Given the above, it is concluded that FBCO can solve difficult optimization problems with the help of fuzzy logic to improve performance. The proposed algorithm has been proven to have better performance in comparison with the traditional BCO method in most of the problems used in this work.

## References

- [1] O. Abedinia, B. Wyns, A. Ghasemi, Robust fuzzy PSS design using ABC, in: 2011 10th International Conference on Environment and Electrical Engineering (EEEIC), IEEE, 2011, pp. 1–4.
- [2] A. Aceves, J. Aguilar, A simplified version of Mamdani's fuzzy controller: the natural logic controller, *IEEE Trans. Fuzzy Syst.* 14 (1) (2006) 16–30.
- [3] L. Amador-Angulo, O. Castillo, Optimization of the Type-1 and Type-2 fuzzy controller design for the water tank using the Bee Colony Optimization, in: 2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW), IEEE, 2014, pp. 1–8.
- [4] L. Amador-Angulo, O. Castillo, Comparison of the optimal design of fuzzy controllers for the water tank using ant colony optimization, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer International Publishing, 2014, pp. 255–273.
- [5] L. Amador-Angulo, O. Castillo, M. Pulido, Comparison of fuzzy controllers for the water tank with Type-1 and Type-2 fuzzy logic, in: IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint, IEEE, 2013, pp. 1062–1067.
- [6] A. Baykasoglu, L. Özbakır, P. Tapkan, Artificial bee colony algorithm and its application to generalized assignment problem, in: T.S.C. Felix, K.T. Manoj (Eds.), *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, Itech Education and Publishing, Vienna, Austria, 2007, pp. 113–143.
- [7] B. Bhushan, S.K. Valluru, M. Singh, Takagi-Sugeno fuzzy system based stable direct adaptive control of nonlinear systems, *Int. J. Comput. Appl.* 68 (15) (2013) 30–36.
- [8] C. Caraveo, O. Castillo, Optimization of fuzzy controllers design using the bee colony algorithm, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer International Publishing, 2014, pp. 163–175.
- [9] O. Castillo, R. Martínez-Marroquín, P. Melin, F. Valdez, J. Soria, Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot, *Inf. Sci.* 192 (2012) 19–38.
- [10] O. Castillo, H. Neyoy, J. Soria, M. García, F. Valdez, Dynamic fuzzy logic parameter tuning for ACO and its application in the fuzzy logic control of an autonomous mobile robot, *Int. J. Adv. Robot. Syst.* (2013) 10.
- [11] T. Chaiyatham, I. Ngamroo, A bee colony optimization based-fuzzy logic-pid control design of electrolyzer for microgrid stabilization, *Int. J. Innov. Comput. Inf. Control* 8 (9) (2012) 6049–6066.
- [12] T. Chaiyatham, I. Ngamroo, S. Pothiya, S. Vachirasricirikul, Design of optimal fuzzy logic-PID controller using bee colony optimization for frequency control in an isolated wind-diesel system, in: *Transmission & Distribution Conference & Exposition: Asia and Pacific*, 2009, IEEE, 2009, pp. 1–4.
- [13] C.S. Chong, M.Y.H. Low, A.I. Sivakumar, K.L. Gay, A bee colony optimization algorithm to job shop scheduling simulation, in: L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, R.M. Fujimoto (Eds.), *Proceedings of the Winter Conference*, Washington, DC, 2006, pp. 1954–1961.
- [14] R. Fierro, O. Castillo, Design of Fuzzy Different PSO Variants, *Recent Advances on Hybrid Intelligent Systems*, Springer, 2013, pp. 81–88.
- [15] J.I. Flores Mendoza, E. Mezura Montes, Dynamic adaptation and multiobjective concepts in a particle swarm optimizer for constrained optimization, in: CEC 2008 (IEEE World Congress on Computational Intelligence) IEEE Congress on Evolutionary Computation, 2008, IEEE, 2008, pp. 3427–3434.
- [16] T.P. Hong, Y.F. Tung, S.L. Wang, M.T. Wu, Y.L. Wu, Extracting membership functions in fuzzy data mining by ant colony systems, in: *International Conference on Machine Learning and Cybernetics*, vol. 7, 2008, pp. 3979–3984.
- [17] A.N. Isizoh, S.O. Okide, A.E. Anazia, C.D. Ogu, Temperature control system using fuzzy logic technique, *Int. J. Adv. Res. Artif. Intell. (IJARAI)* 1 (3) (2012).
- [18] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-fuzzy and soft computing – a computational approach to learning and machine intelligence, *IEEE Trans. Autom. Control* 42 (10) (1997) 1482–1484.
- [19] D. Karaboga, An Idea based on Honey Bee Swarm for Numerical Optimization (Technical Report-Tr06, October, 2005), Erciyes University, Engineering Faculty Computer Engineering Department, Kayseri/Türkiye, 2005.
- [20] D. Karaboga, B. Basturk, On the performance of Artificial Bee Colony (ABC) algorithm, *Appl. Soft Comput.* 8 (2) (2008) 687–697.
- [21] D. Karaboga, B. Akay, C. Ozturk, Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks, Springer Verlag, Berlin, Heidelberg, 2007, pp. 318–329.
- [22] Z. Li, C. Xu, Adaptive fuzzy logic control of dynamic balance and motion for wheeled inverted pendulums, *Fuzzy Sets Syst.* 160 (12) (2009) 1787–1803.
- [23] P. Lučić, D. Teodorović, Transportation modeling: an artificial life approach, in: *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, Washington, DC, 2002, pp. 216–223.
- [24] P. Lučić, D. Teodorović, Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach, in: J.L. Verdegay (Ed.), *Fuzzy Sets in Optimization*, Springer-Verlag, Heidelberg, Berlin, 2003, pp. 67–82.
- [25] K.F. Man, K.S. Tang, S. Kwong, *Genetic Algorithms: Concepts and Designs*, Springer, 2000.
- [26] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, M. Valdez, Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic, *Expert Syst. Appl.* 40 (8) (2013) 3196–3206.
- [27] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems, Introduction and New Directions*, PH PTR, 2001.
- [28] H. Neyoy, O. Castillo, J. Soria, Dynamic fuzzy logic parameter tuning for ACO and its application in TSP problems, in: *Recent Advances on Hybrid Intelligent Systems*, Berlin, Heidelberg, Springer, 2013, pp. 259–271.
- [29] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans. Evol. Comput.* 13 (2) (2009) 398–417.
- [30] D. Teodorović, M. Dell'Orco, Bee colony optimization – a cooperative learning approach to complex transportation problems, in: *Advanced OR and AI Methods in Transportation*. Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland, 2005, pp. 51–60.
- [31] D. Teodorović, P. Lučić, G. Marković, M. Dell'Orco, Bee colony optimization: principles and applications, in: B. Reljin, S. Stanković (Eds.), *Proceedings of the Eight Seminar on Neural Network Applications in Electrical Engineering – EUREL 2006*, University of Belgrade, Belgrade, 2006, pp. 151–156.
- [32] D. Teodorović, M. Šelmić, The BCO algorithm for the p median problem, in: *Proceedings of the XXXIV Serbian Operations Research Conference*, Zlatibor, Serbia, 2007 (in Serbian).
- [33] D. Teodorović, *Swarm intelligent systems for transportation engineering: principles and applications*, *Transp. Res. Part C: Emerg. Technol.* 16 (2008) 651–782.
- [34] Teodorovic, Bee colony optimization (BCO), in: C.P. Lim, L.C. Jain, S. Dehuri (Eds.), *Innovations in Swarm Intelligence*, Springer-Verlag, 2009, pp. 39–60.
- [35] The MathWorks, *Fuzzy Logic Toolbox For use with Matlab. User's Guide Version 2*, 2006.
- [36] J. Vesterstrom, R. Thomsen, A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, in: CEC2004 Congress on Evolutionary Computation, 2004, vol. 2, IEEE, 2004, pp. 1980–1987.
- [37] L.P. Wong, M.Y.H. Low, C.S. Chong, Bee colony optimization with local search for traveling salesman problem, in: *Proceedings of the 6th IEEE International Conference on Industrial Informatics*, 2008, pp. 1019–1025.
- [38] L. Zadeh, Fuzzy logic, *IEEE Comput.* (1965) 83–92.
- [39] A.A. Zamani, E. Bijami, F. Sheikholeslam, B. Jafarsteh, Optimal fuzzy load frequency controller with simultaneous auto-tuned membership functions and fuzzy control rules, *Turk. J. Electr. Eng. Comput. Sci.* 22 (1) (2014) 66–86.