Information Sciences 294 (2015) 203–215

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system

Oscar Castillo *, Evelia Lizárraga, Jose Soria, Patricia Melin, Fevrier Valdez

Tijuana Institute of Technology, Tijuana, Mexico

ARTICLE INFO

Article history:
Received 20 May 2013
Received in revised form 10 September 2014
Accepted 24 September 2014
Available online 2 October 2014

Keywords:
Ant colony optimization
Fuzzy logic
Fuzzy control design

ABSTRACT

In this paper we describe the design of a fuzzy logic controller for the ball and beam system using a modified Ant Colony Optimization (ACO) method for optimizing the type of membership functions, the parameters of the membership functions and the fuzzy rules. This is achieved by applying a systematic and hierarchical optimization approach modifying the conventional ACO algorithm using an ant set partition strategy. The simulation results show that the proposed algorithm achieves better results than the classical ACO algorithm for the design of the fuzzy controller.

1. Introduction

Recently it has been shown that fuzzy logic controllers (FLC) provide a good alternative to conventional control systems [3,16,34]. However, a FLC must be designed specifically for a given system; that is, its parameters must be adjusted *ad hoc*. This adjustment is usually performed "manually" or by trial and error [21]. Since this usually takes a lot of time for an FLC designer, several bio-inspired and evolutionary techniques have been proposed for this purpose, such as: Genetic Algorithms (GAs) [2], Particle Swarm Optimization (PSO) [10] and Ant Colony Optimization (ACO) [9], among others; the latter approach being the one that is considered in this paper.

ACO algorithms have been successfully applied to versatile combinational optimization problems, namely vehicle routing [23], quadratic assignment problem (QAP) [32], job-shop scheduling [13], the TSP problem [31] and the ball and beam system [22]. The former being our case of study which is an important and classic laboratory model for teaching engineering and control systems in the research area. Because it is a simple and easy to understand system, it can be used to study many classic and modern design methods in control engineering as it has a very interesting property for the control engineer: it is unstable in open loop.

In this paper we describe the proposed methodology to design a fuzzy system using a new approach based on the ACO metaheuristic. The proposed algorithm modifies the classical ACO algorithm, i.e. a new modification of the algorithm has been proposed to improve the efficiency and accuracy. To explain this in more detail, the proposed algorithm is viewed as a series of steps that are performed hierarchically and sequentially, allowing a faster optimization and better results, compared with the classical ACO algorithm [1].

* Corresponding author.
E-mail address: ocastillo@tectijuana.mx (O. Castillo).

http://dx.doi.org/10.1016/j.ins.2014.09.040

The proposed algorithm performs the overall optimization by dividing the total number of ants, equivalently, among the five main ACO variants [26,29]: Ant System (AS), Elitist Ant System (EAS), Rank Based Ant System (ASRank), Man-Min Ant System (MMAS), Ant Colony System (ACS). To improve the performance of the proposed method a stagnation mechanism was added that would stop a particular variant that is not producing optimal results. This approach has been applied previously for the Traveling Salesman Problem (TSP) providing satisfactory results [19,24]. In other works, several ACO variants have been tried simultaneously [3,13], but not for designing fuzzy controllers and with the ant set partition strategy, which is the proposed contribution in this paper.

We performed several experiments applying the proposed method to the ball and beam problem, whose objective is to stabilize the balance of the ball using a fuzzy logic controller to determine the new position of the ball according the angle of the beam [22]. The proposed ACO algorithm optimizes the type of membership functions, parameters of the membership functions and fuzzy rules of the FLC.

In Naredo and Castillo [22] the same 5 variants of ACO were applied in an individual fashion to design fuzzy controllers for the ball and beam problem and at the end one of the variants was selected as the best one for this benchmark control problem. In particular the AS variant was determined to be the best one for this problem. In the present paper we are now considering the use of the 5 variants simultaneously to improve the performance with the ant set partition strategy and its application to the ball and beam benchmark problem.

Chang et al. [3] proposed an algorithm with the capability to update the pheromone, adaptive parameter tuning, and mechanism resetting. The proposed method is utilized to tune the parameters of the fuzzy controller for a real beam and ball system. The control of the ball and beam system is decoupled into two subsystems, the position control of ball and the balance control of beam. Two unique fuzzy control strategies are utilized to balance the beam and to keep the ball in the designated position. The proposed fuzzy ACO optimized control scheme contains a fuzzy beam-balance controller, a fuzzy ball-position controller, and the fuzzy ACO tuning mechanism.

Other related work is the one by Martinez et al. [20] in which both ACO and genetic algorithms (GAs) are compared for fuzzy controller design showing that ACO could outperform GAs in this kind of problems. However, the benchmark control problem is not the ball and beam and no direct comparison of results could be made with the work presented in this paper. We have to say that only the simple ACO was used in the work presented in [14] and no modification of the ACO was proposed.

This paper is organized as follows. Section 2 describes the basic concepts of ant colony optimization. Section 3 details the proposed approach of ant partition and the problem of graphic representation. Section 4 describes the Ball and Beam system and the fuzzy logic controller to stabilize the system. Section 5 shows the simulations results. Section 6 presents a statistical test for the proposed approach. In Section 7, a conclusion of this study is presented.

2. Basic concepts of ant colony optimization

Proposed as an ant colony analogy by Dorigo in 1990s [9], ACO is defined biologically speaking by ants who aim to find food visiting potential food places. To communicate, ants use stigmergy, which is a biological mechanism that can transmit information through the environment by using pheromones exuded by ants. Artificially speaking, each ant is a possible solution to a problem, where the set of possible solutions is represented as a graph [5–8].

In the ACO algorithm an ant k visits each node. To select the next node j, a stochastic probabilistic rule (1) is applied, which is determined by using information of the amount of pheromone τ_{ij} in node i, within a feasible neighborhood N_i^k,

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

The heuristic information η_{ij} is provided by the Root Mean Squared Error (RMSE) given by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{q=1}^{N}(u_q - \tilde{u}_q)^2}, \tag{2}$$

where $\hat{u}$ is the control signal, u the reference and N the number of observed points, and the parameters α and β are the relative weights of pheromone and heuristic information, respectively.

Once the path is constructed it will be evaluated to determine the cost of the path. Depending on the cost of the path is the amount of pheromone $\Delta\tau_{ij}$ that an ant will deposit on the node (3). The better the path, the larger the amount of pheromone that will be deposited by the ants and this is represented by:

$$\tau_{ij} = \rho\tau_{ij} + \Delta\tau_{ij}, \tag{3}$$

where ρ is a parameter that represents the evaporation coefficient, $0 < \rho < 1$.

The algorithm terminates when the path created by each ant has been evaluated.

The general steps of the basic ACO algorithm are the following [11,12]:

1. Set a pheromone concentration τ_{ij} to each link (i,j).
2. Place a number $k = 1, 2, \ldots, n_k$ in the nest.
3. Iteratively build a path to the food source (destiny node), using Eq. (2) for every ant.
   - Remove cycles and compute each route weight $f(x^k(t))$. A cycle could be generated when there are no feasible candidate nodes, that is, for any i and any k, $N_i^k = \emptyset$; then the predecessor of that node is included as a former node of the path.
4. Apply evaporation using Eq. (3).
5. Update the pheromone concentration using Eq. (3)
6. Finally, finish the algorithm in any of the three different ways:
   - When a maximum number of epochs have been reached.
   - When it has to be an acceptable solution, with $f(x_k(t)) < \varepsilon$.
   - When all ants follow the same path.

3. Proposed optimization method: ACO Variants Subset Evaluation (AVSE)

Having a set of m ants the method equivalently divides the total number of ants in five different subsets and each one is evaluated separately by the corresponding variation of ACO (AS, EAS, ASRank, ACS and MMAS) and Fig. 1 illustrates this approach. The evaluation of the each variant (ant subset) is made sequentially in the same iteration i. Subsequently, the best ant of each partition is compared with each other; obtaining the best global ant i (global best i) as illustrated in Fig. 2.

Therefore, in each iteration the best global ant i is compared to the best global ant $i-1$ following the conventional ACO algorithm. This allows us to compare different variations per iteration, saving time in doing tests and having fewer overheads in comparison with the conventional method, which uses just one variation of ACO in all iterations. This means n (iterations) $\times$ 5 (variations) number of tests comparison with the proposed method which only needs n tests. Because the proposed method selects one ant per iteration that represents also a variant of ACO; the end result provides the most used variants, hence the one with the best performance.

To improve the performance of the proposed AVSE method, we added a stagnation mechanism, which allows us to stop a variant that has not been giving good results after 5 consecutive iterations. It is understood as a good result if a variant reaches a local best, per iteration. It has been noted in several preliminary experiments that if after an average of 5 iterations a variant does not reach a local best is usually stagnating. The use of this mechanism allows a faster and proficient performance, using only the variants that obtain better results.

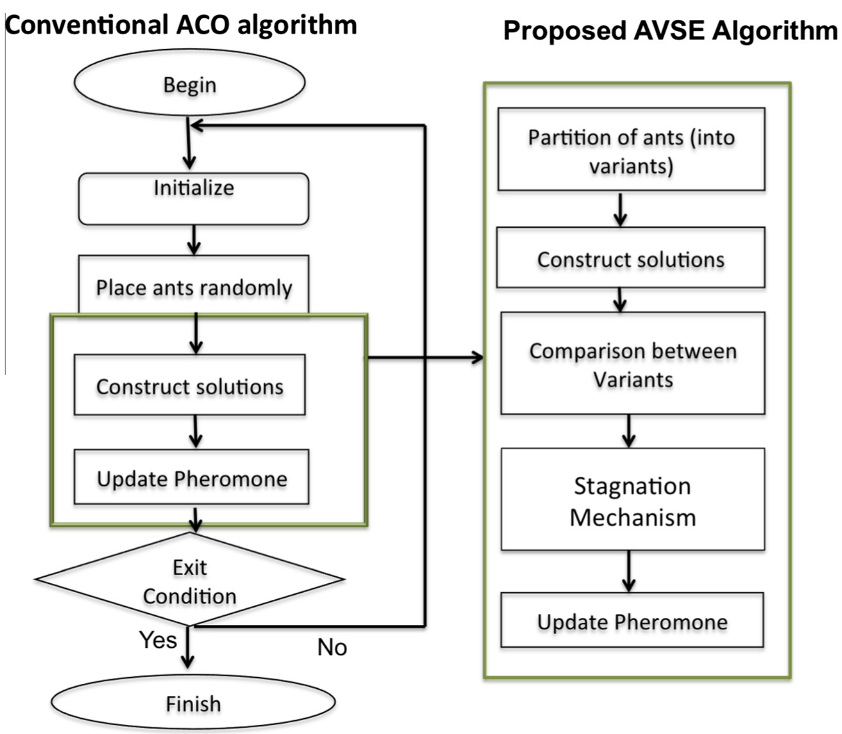

Fig. 1. Proposed AVSE algorithm.

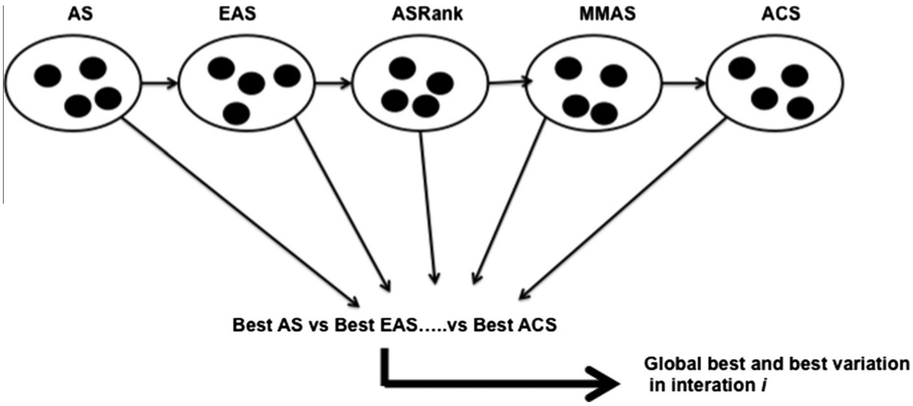

Fig. 2. Ants evaluation in subsets.

3.1. Hierarchical optimization

Because the proposed optimization is performed in the three main parts of the fuzzy system (type, parameters and rules) we must decide which comes first. Thus our optimization is sequential and hierarchical. Sequential, because is made successively, and not parallel; using the same number of iterations and ants for each kind of optimization. It is hierarchical, because we optimize first, the types of membership functions, then parameters of membership functions and finally, the optimization of fuzzy rules is performed. The justification for this strategy is that in order to optimize parameters, we need to know what types of membership function the fuzzy system has. Optimization of fuzzy rules was done at the end, as it is the one that takes less time and does not affect in the other kinds of optimizations (type and parameters). Fig. 3 illustrates how the optimization sequence is structured.

First, the AVSE algorithm optimizes the type of membership functions; when a best global is reached, the algorithm saves the best global as *type-global-best* and also saves the best fuzzy system obtained as *type-best-fuzzy*. Once the maximum number of iterations is completed, the algorithm continues with the optimization of parameters of the membership functions

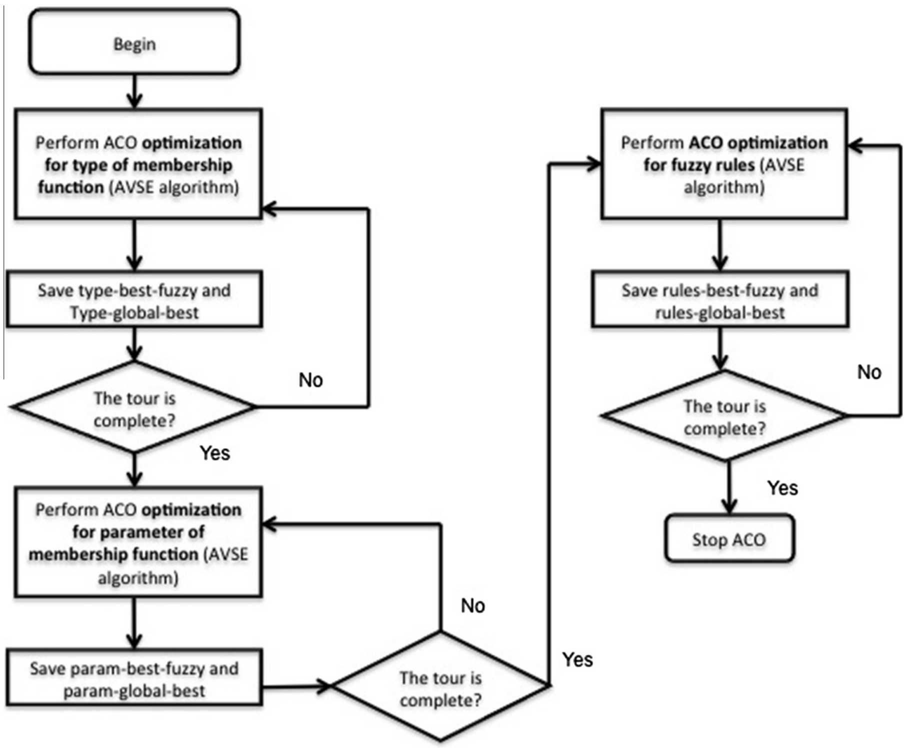

Fig. 3. Hierarchical sequence for AVSE optimization.

setting the same maximum of iterations and ants, using the best resulting fuzzy system obtained in the previous optimization (*type-best-fuzzy*), it also sets the global best as *type-global-best*. Similarly, when performing parameter optimization the algorithm will save the best fuzzy system found as *param-best-fuzzy* and the global best as *param-global-best*. As in the previous optimization, the methodology for optimizing fuzzy rules is similar; saving the best fuzzy system as *rules-best-fuzzy,* and if reached, the *rules-global-best*. At the end we obtain, the best fuzzy system found, which is the final result after performing the three types of optimization mentioned above.

3.2. Graphic representation

3.2.1. Automatic parameter extraction

One of the main contributions of this method is to generalize the optimization of fuzzy systems. To this end, we have developed two graphical interfaces. The first one is called Fuzzy Parameter Extraction (FPE), which allows the extraction of the fuzzy parameters needed to construct the pheromone matrix, such as:

- Total number of inputs.
- Total number of outputs.
- Total number of MF per input and output.
- Type of membership function per input and output.
- Range per input and output.

In the above mentioned interface, the user loads the fuzzy system to be optimized and the corresponding plant, and also can select the type of optimization that will be performed, choosing at least one. This will make the optimization automatic, regardless of the type of fuzzy system, the number of inputs or outputs, or its membership function types, allowing freedom and easiness to perform the experiments.

The second interface is called Pheromone Matrix Constructor, and it is used to input the necessary ACO parameters to construct the pheromone matrix and the solution nodes. The user enters the following data:

- Total number of ants (that will be divided in n subsets of ants). Where n is the number of variations selected.
- Maximum number of iterations.
- Alpha.
- Beta.
- Rho (evaporation rate).
- Variations to use in optimization (AS, EAS, ASRank, MMAS, ACS) with a minimum of two.
- Depending of the chosen variation, the user can introduce other parameters; the e constant for the Elitist Ant System and the w constant for the Rank-Based Ant System.

Because of the nature of the ACO algorithm, the optimization problem has to be represented as a graph. This requires representing all the possible solutions for each type of optimization (parameters, type and rules) and each type of optimization will have a different graph representation. Each resulting graph will generate a three-dimensional matrix, which will vary in dimension.

3.2.2. Graph representation of type of membership function optimization

Rows: as was defined before, we will use the 4 main types of membership functions; this is reflected as 4 rows for the pheromone matrix.

Columns: this depends on the number of membership functions per input and output, which is information provided by the FPE interface. For the Ball and Beam problem the total number of MFs per input is 8. Because the Ball and Beam fuzzy system has the Takagi–Sugeno–Kang (TSK) architecture there is no need to optimize the type of membership function, given that the outputs are mathematical equations.

Third dimension: this will be given by the number of ants in each subset, not the total of ants. For example if the total number of ants are 150 and we use three variations to optimize, the third dimension of the matrix will be 150/3 = 50 ants.

The resulting graph for the Ball and Beam problem is illustrated in Fig. 4.

3.2.3. Graph representation of parameters of membership function optimization

As for the optimization of membership function types, the universe of discourse can be represented as a graph, where the rows of the pheromone matrix are the number of possible values for each parameter. Because the range values in the inputs are not the same, we decided to discretize the possible values in 100 parts and normalize the values from 0.01 to 1. So this means that the pheromone matrix will have 100 rows.

The columns will be the total number of parameters to be optimized. Because the points to be moved (parameters) depend on the type and location of a MF, we established a general way to decide which points are going to be optimized using Fig. 5 as a pattern. Where the solid circle is the fixed point (a point that will not be optimized) and the empty circle represents the point to be optimized.

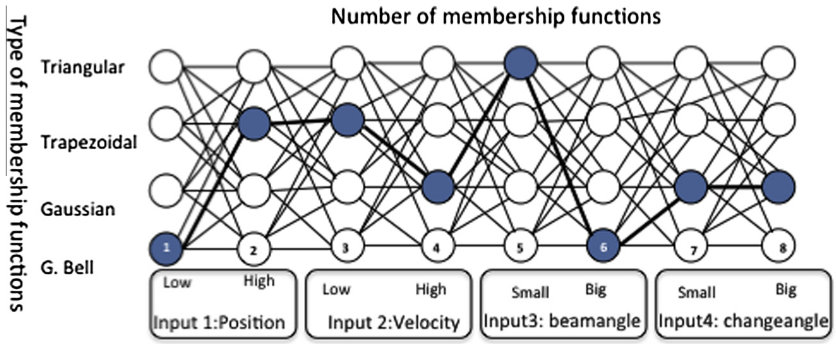

Fig. 4. Graph representation of type of membership function optimization.

As a note, a fuzzy system requires that the MFs overlap with each other; this is why our method has an internal validation of points. Of course, before we can optimize the parameters we need to have what type of membership functions will result in the *type of membership function optimization.* For the first kind of optimization (type of membership function) we used as a base the non-optimized Ball and Beam FLC, which has as a generalized bell type in all the membership functions for all the inputs. So the resulting columns for the Ball and Beam FLC are as shown in Fig. 6.

Finally, the third dimension of the pheromone matrix will be the number of ants as calculated on the type of membership function optimization. Fig. 7 shows the resulting representation for the Ball and Beam problem, where m is the total number of ants.

3.2.4. Graph representation of fuzzy rules optimization

To perform the optimization in the fuzzy rules, we have to consider how many antecedents are needed (number of membership functions). As in the previous optimizations, it is necessary to represents the possible combinations in a graph, making the artificial ant build a tour, and each tour will be a set of fuzzy rules.

The basic actions of this part of the algorithm are as follows:

(1) Combine antecedents.
(2) Assign a consequent to each combination of antecedents of the rules.
(3) Activate the necessary rules.

The first step is to create the graph for all the possible combinations of X antecedents and consequent Y as follows:

If $X_{1,1}$ and $X_{1,2}$ and $X_{1,3},\ldots,X_{nj}$ then $Y_{1,1}$
If $X_{2,1}$ and $X_{2,2}$ and $X_{2,3},\ldots,X_{nj}$ then, $Y_{1,2}$
If $X_{3,1}$ and $X_{3,2}$ and $X_{3,3},\ldots,X_{nj}$ then $Y_{1,3}$
$\ldots$
If $X_{1,J}$ and $X_{2,j}$ and $X_{3,j}$ then $X_{n,j}$ then $Y_{o,h}$,

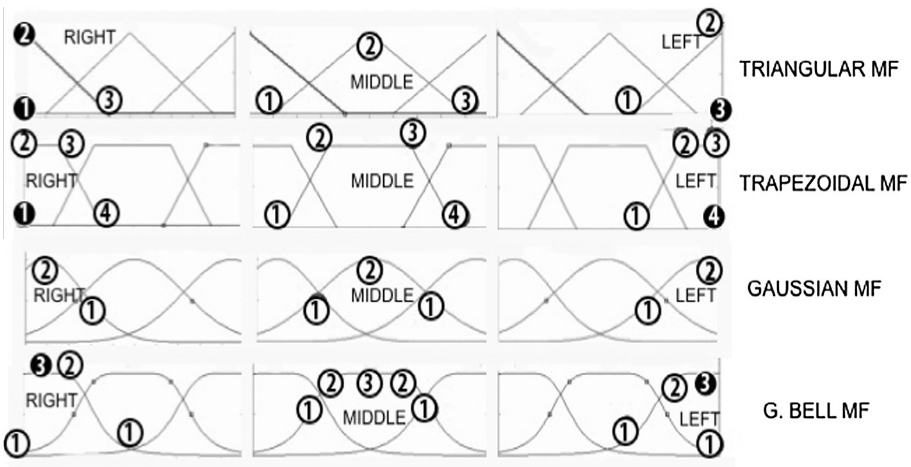

Fig. 5. Relation between type and location of a membership function.

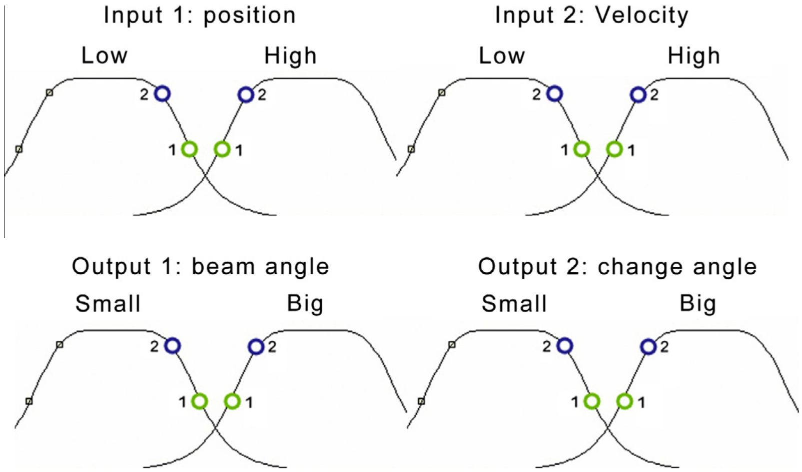

Fig. 6. Points to be optimized in the Ball and Beam FLC.

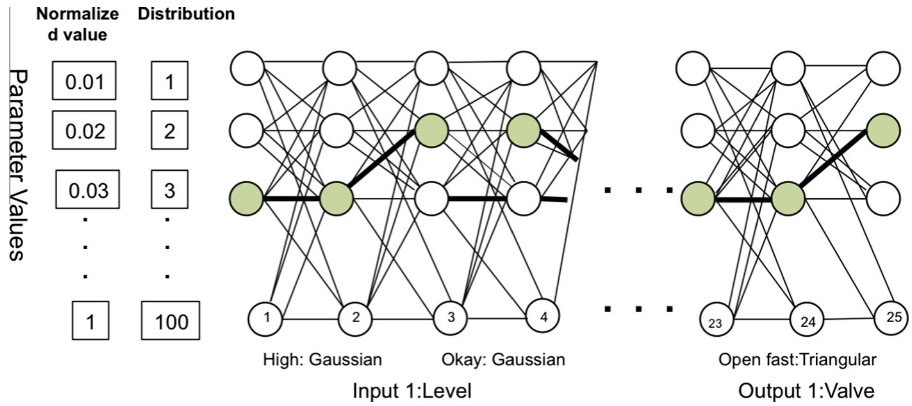

Fig. 7. Graph representation of pheromone matrix for the parameter optimization.

where *i* is the number of inputs, *j* are the linguistic labels (membership functions) per input, *o* is the number of outputs and *h* are the linguistic labels per output [25].

Table 1 shows a representation of how we can combine antecedents and consequents, where the highlighted cells with the number 1 are the activated antecedents.

The construction of the rules is performed by considering row by row from top to bottom as Fig. 8 illustrates:

Rule 1: If input_1 is $\mathbf{MF}_{1,1}$ and input_2 is $\mathbf{MF}_{2,1}$, then output_1 is $\mathbf{OMF}_{1,1}$.

Rule 2: If input_1 is $\mathbf{MF}_{1,1}$ and input_2 is $\mathbf{MF}_{2,2}$, then output_1 is $\mathbf{OMF}_{1,1}$.

Table 1
Combination of antecedent and consequents.

	ANTECEDENTS					
	Input_1			Input_2		
	$\text{IMF}_{1,1}$	$\text{IMF}_{1,2}$	$\text{IMF}_{1,j}$	$\text{IMF}_{2,1}$	$\text{IMF}_{2,2}$	$\text{IMF}_{i,n}$
CONSEQUENTS						
Output_o						
$\text{OMF}_{1,1}$	**1**	0	0	**1**	**1**	**0**
$\text{OMF}_{1,2}$	0	**1**	0	0	0	**0**
$\text{OMF}_{1,3}$	**0**	0	0	0	**1**	0
$\text{OMF}_{1,4}$	0	0	**0**	0	0	0
$\text{OMF}_{o,h}$	0	0	**1**	0	0	0

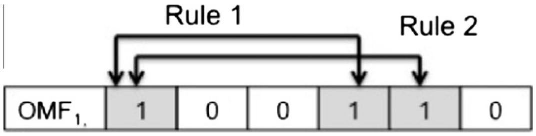

Fig. 8. Rule constructions according to activated cells.

4. Ball and beam fuzzy logic controller

The used benchmark control system is illustrated in Fig. 9, which is implemented by a steel ball that rolls on a bar, mounted on the shaft of an electric motor [22].

The steel ball, in this configuration, can be tied to its axis by applying an electrical control signal to the motor amplifier. The goal of control is to automatically adjust the position of the ball, shifting the angle of the bar.

4.1. Mathematical model

The force that accelerates the ball, while it rolls, comes from a gravitational acceleration, which acts parallel to the bar. The horizontal line is the force equivalent to: $mgsin\ \theta$.

Thus, the equation regarding force = (mass)(acceleration), can be simplified as follows:

$$mgsin\theta = m\ddot{x}, \tag{4}$$

where m is the mass of the ball, g is the gravitational constant, θ is the angle, and x is the position of the ball on the bar.

Because small angles are required, $sin\theta$ is approximately equal to θ, then:

$$\ddot{x} = g\theta. \tag{5}$$

In which θ is proportional to the angle controlled by motor voltage u, and the position x is detected by sensor y.

By replacing θ for the control voltage u, and the position x of the ball by the position measured by the sensor y, it is possible to find a single constant b that represents the total gain in the response from the control voltage:

$$\ddot{y} = bu. \tag{6}$$

Therefore the transfer function can be obtained as follows:

$$x(s) = \frac{b}{s^2}u(s). \tag{7}$$

Since the system behaves in an unstable fashion from the beginning because there is an open loop, it is necessary to measure the speed of the ball constantly. This can be done using an "observer" to estimate the system states and use such estimates for position and velocity; to do so, it is usually proposed a state feedback controller, in this case a fuzzy controller.

The ball and beam system was implemented using the following parameters:

- Mass of the Ball (M) = 0.11 kg.
- Radius of the Ball (R) = 0.015 m.
- Lever arm offset (d) = 0.03 m.
- Gravitational acceleration (g) = 9.8 m/s^2.
- Length of the beam (L) = 1.0 m.
- Ball's moment of inertia (J) = 9.99e−6 kg m^2.

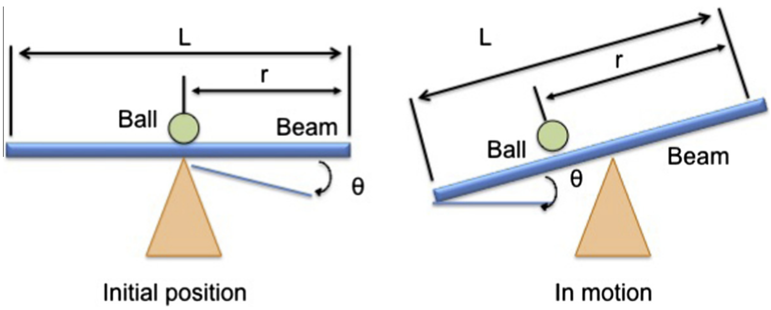

Fig. 9. Ball and Beam system.

- Ball position coordinate (r).
- Beam angle coordinate (α).
- Servo gear angle (θ).

We defined a ball placed on a bar, where it can roll with one degree of freedom, and a lifting arm attached to the bar at one end and a servomotor in another. When the angles switch into a horizontal position, the gravity will cause the ball to rollover. To solve this, we designed a controller for the system so that the position of the ball can be manipulated.

For this problem the sliding and the friction between the bar and the ball are considered negligible. The Lagrangian equation for motion of the ball is given by:

$$0 = \left(\frac{J}{R^2} + m\right) r + mg sin\ddot{\alpha} - mr(\alpha)^2. \tag{8}$$

The beam angle (α) can be expressed according to the angle of the gear (θ),

$$\alpha = \frac{d}{L}\theta. \tag{9}$$

The fuzzy logic controller was constructed according to the position of the ball, velocity and beam angle (inputs), to determine the position of the ball (output). The resulting structure of the Sugeno fuzzy logic controller is shown in Fig. 10

The fuzzy rules that form the Sugeno fuzzy system in Fig. 10 contain the basic knowledge about controlling the ball and beam benchmark plant. Basically, the new position of the ball depends on 4 input variables, which are the previous position, the velocity, the beam angle and the change of the angle. Of course, the optimal design of this fuzzy controller is performed with the ACO approach proposed in the paper.

5. Simulations results

In this section we present the results of the optimized ball and beam FLC using AVSE. Each table of results displays the type of optimization performed (parameters, type of membership function and fuzzy rules), variant that was used, number of experiments and AGB which stands for Average Global Best, given by (10), where GB is the global best per iteration and "*it*" is the maximum number of iterations.

$$AGB = \frac{GB}{\sum_{i=1}^{it} GB_i}. \tag{10}$$

EGB stands for Experiment Global Best calculated using (11), where n is the total number of experiments (30),

$$EGB = \frac{AGB}{\sum_{x=1}^{n} AGB_x}. \tag{11}$$

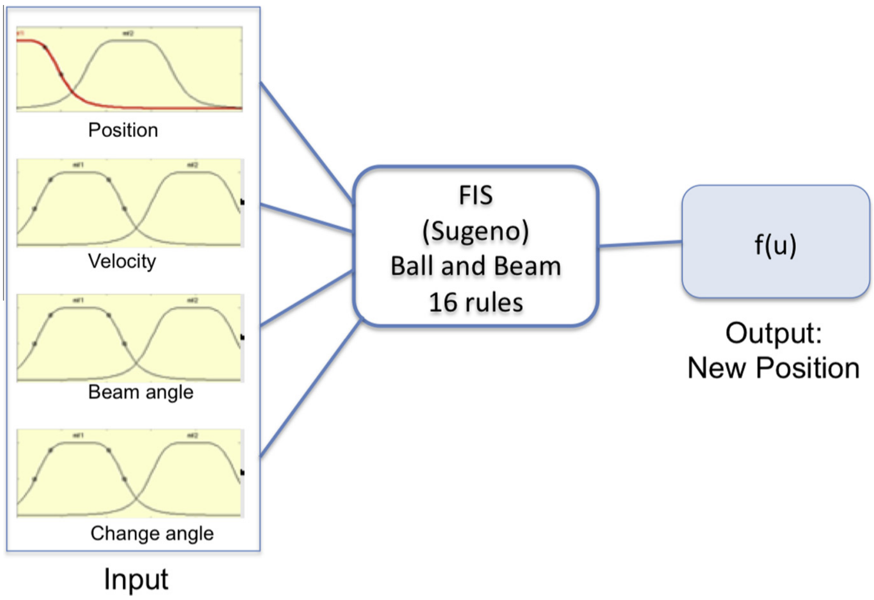

Fig. 10. Ball and Beam FLC.

Table 2
Combination of different types of optimization using only AVSE.

Optimization type	Variant	Experiments	AGB	EGB
Type of MF	AVSE	10	0.0045	0.0092
Parameters	AVSE	10	0.0039	0.008
Fuzzy Rules	AVSE	10	0.0714	0.012
Type MF + Parameters	AVSE	10	0.00162	0.0067
Type MF + Fuzzy Rules	AVSE	10	0.020	0.059
Parameters + Fuzzy Rules	AVSE	10	0.0042	0.090

Table 2 combines types of optimization using only AVSE, and Table 3 compares the performance between each variant individually and the proposed AVSE method using the three kinds of optimization. Table 4 compares a non-optimized FLC with AVSE using a particular reference. Table 5 illustrates the comparisons between AVSE and a related work [17], which only performed the type of membership optimization and parameters optimization. Finally, a statistical test is presented to verify that the proposed method is significantly better than the one presented in [17].

For Table 2, we used the same parameters, performing 10 experiments. For Tables 3 and 4 we performed 30 experiments with the proposed AVSE method and 30 experiments using each variant independently. For Table 5 we compared our AGB with another approach for an ACO optimizing a FLC for the same problem.

The values of the parameters are: alpha = 1, beta = 3, rho = 0.7, w = 6, e = 6 for the optimization of type and e = 1.1 for the optimization of parameters. We used these parameter values according to the literature [2,3]. We tested several values within the recommended range. The number of iterations was defined as100 and the number of ants is 50, and these values were chosen based on previous experiments.

These experiments represented in Table 2 were performed to compare which kind of optimization is more relevant and then, based on this, perform a "diagnosis" of the system. As a diagnosis, it is meant to be an informal demonstration to prove if just one kind of optimization is necessary to produce a good result or if it is the combination of two or more types of optimization that provides a considerable difference in the end. Indeed, we can observe that when the types of membership functions and parameters are optimized, the error is smaller. This can lead to an interesting analysis of the system, and not only for the ball and beam case. This means that the type of membership functions and parameters are fundamental for a good performance of an FLC. Our explanation is, that the type of membership functions determines the amount of maximum and minimum values, and parameters determine the range of the mentioned values; for example, a triangular

Table 3
Comparison between AVSE with each variant combining types of optimization.

Optimization type	Variant	Experiments	AGB	EGB	Time (Hours)
Type + Parameters + Rules	AVSE (AS, ACS, EAS, MMAS, ASRank) Winner MMAS	30	0.00156	0.00577	10:11
Type + Parameters + Rules	AS	30	0.0322	0.0371	3:11
Type + Parameters + Rules	ACS	30	0.00243	0.00876	5:42
Type + Parameters + Rules	EAS	30	0.00488	0.00913	4:39
Type + Parameters + Rules	ASRank	30	0.00853	0.01225	2:16
Type + Parameters + Rules	MMAS	30	0.00211	0.00734	2:58

Table 4
Comparison between AVSE with a non-optimized FLC for the Ball and Beam.

Optimization type	Variant	Experiments	AGB	EGB
Type MF + Parameters + Fuzzy Rules	AVSE	30	0.00156	0.00577
NA	NA	30	0.089	0.89

Table 5
Comparison between AVSE with Ref. [22].

Optimization type	Variant	Experiments	AGB
Type MF + Parameters + Fuzzy Rules	AVSE Winner MMAS	5	0.00597
Type MF + Parameters	AS	5	0.07521
Type MF + Parameters	ACS	5	0.10189
Type MF + Parameters	EAS	5	0.0881
Type MF + Parameters	ASRank	5	0.0956
Type MF + Parameters	MMAS	5	0.1113

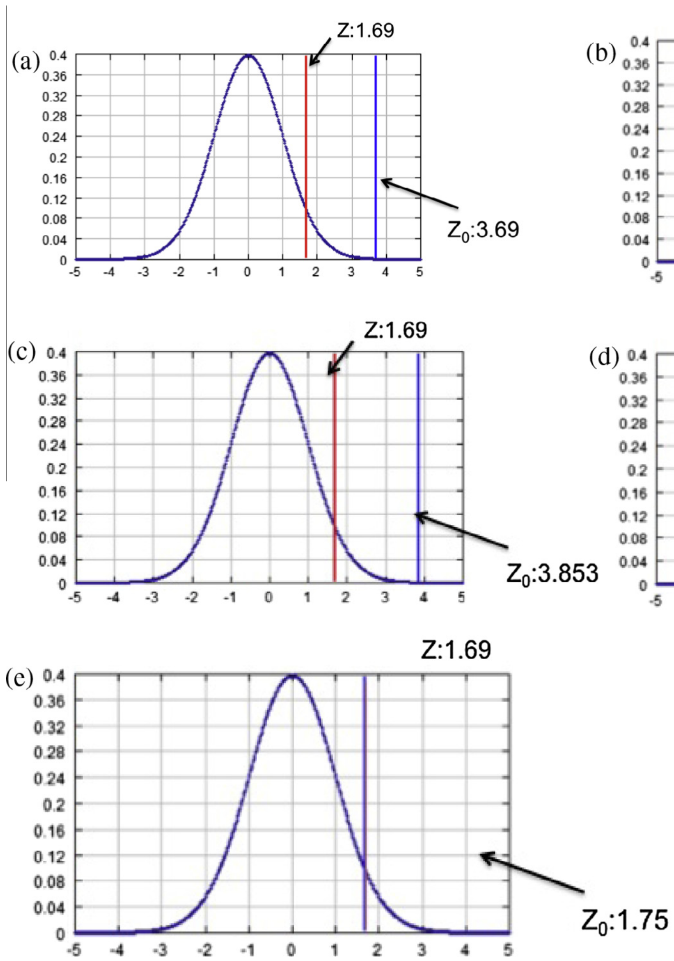

Fig. 11. Results of the statistical hypothesis testing performed for (a) AS vs. AVSE, (b) ASRank vs. AVSE, (c) ACS vs. AVSE, (d) EAS vs. AVSE, (e) MMAS vs. AVSE.

membership function has fewer values with maximum points than a trapezoidal membership function, and as consequence, the range of values that a membership function can have is restricted.

Table 3 shows the results when we optimize the ball and beam FLC using type of membership functions, parameters and fuzzy rules.

As we predicted, when we perform the three kinds of optimizations, the error is smaller, but not that significantly if compared where we have not optimized fuzzy rules. This implies that fuzzy rules optimization is not necessary in this case; as a note, we cannot generalize this for all fuzzy logic controllers. However, in comparison, the proposed AVSE algorithm has smaller errors than the ones reached for each of the other variants individually. This is, because the "competition" among the variants in AVSE provides the better variant that "won" in each iteration; meaning, that the AVSE method will always use the best variation, and therefore, will have the best result in each iteration compared to a variation that is not as good. For example we can observe that, apart from AVSE, the best variations are MMAS and ACS. And the "winner" in AVSE is MMAS, which makes sense, because AVSE stops the "weak" variations if they are not performing well after the fifth consecutive iteration, using the stagnation mechanism.

5.1. Control and reference experiment

To show the effectiveness of the proposed AVSE method, we proposed to adjust the parameters using a non-optimized FLC. We performed 30 experiments using a reference as follows: a beam with an angle of -5° and a ball positioned at 20 cm over the beam. The initial states of the ball and beam system are set to be 0. The objective of the FLC is to adjust the parameters, which contain the error constant, the parameters of the input membership functions and the parameters of the output membership function. In this case, the aim of optimization is to minimize the cost function. The cost function is the root mean square error (RMSE), as we defined previously in Eq. (2). The ball and beam system successfully balances the beam to $-5°$ and positions the ball to desired position of 20 cm; we define as a correct response when the ball is stopped as desired and the beam is also at the required angle, respectively. The results of the experiment based on the performance criterion are shown in Table 4.

The experiments in Table 4 were performed to determine how much difference there exists between an optimized FLC to an ''ad hoc'' FLC that is designed by trial and error. The FLC that was used for these experiments, was the same that we used as a base to perform the optimization. The original non-optimized FLC was designed using the standard configuration for the FLC found in the literature [3,32].

The following experiments are a comparison between AVSE and another approach for an ACO optimizing the FLC applied to the Ball and Beam system, using 5 different ACO variants. Given that Ref. [17] used different parameters, we performed our experiments using the same parameters for AVSE: 10 ants, 100 maximum iterations, alpha = 1, beta = 2, rho = 0.1. Table 5 shows a summary of the comparison of the proposed method and the work presented in [22].

As we can observe, our method gives better results than the other approach that only optimizes the type of membership functions and parameters, even when we used other parameters for beta and rho. This is because our stagnation mechanism allowed the MMA variant to keep functioning along the 100iterations, thus causing that the error values were smaller than if we used the remaining variants. It can be noticed that this happens even long after exceeding the 50 iterations.

6. Statistical tests for AVSE

Because in some cases there is still a vague interpretation of the results, a statistical test for AVSE compared with each variant is needed [3]. A Z test was performed for two samples means with a significance level of 5%, to determine whether the proposed method is better than the classical approach. Case 1 compares the null hypothesis $H_0 \mu_{AS} \leqslant \mu_{AVSE}$ with the alternative hypothesis $H_u \mu_{AS} > \mu_{AVSE}$. Case 2 compares the null hypothesis $H_0 \mu_{ASC} \leqslant \mu_{AVSE}$ with the alternative hypothesis $H_u \mu_{AS} > \mu_{AVSE}$. Case 3 compares the null hypothesis $H_0 \mu_{EAS} \leqslant \mu_{AVSE}$ with the alternative hypothesis $H_u \mu_{EAS} > \mu_{AVSE}$. Case 4 compares the null hypothesis $H_0 \mu_{ASRank} \leqslant \mu_{AVSE}$ with the alternative hypothesis $H_u \mu_{ASRank} > \mu_{AVSE}$. Finally, case 5 compares the null hypothesis $H_0 \mu_{MMAS} \leqslant \mu_{AVSE}$ with the alternative hypothesis $H_u \mu_{MMAS} > \mu_{AVSE}$ [4].

No statistical evidence was found with a significance level of 5% such that the averages of AS, ACS, EAS, ASRank or MMAS are greater than the average of AVSE (Fig. 11a–e). On the other hand, we can say that there is statistical evidence that AVSE outperforms the individual variants. However, we can notice in Fig. 11e that both Z's are overlapped when we compared MMAS with AVSE, this is actually a confirmation that AVSE is using MMAS constantly in each iteration, thus the values are almost the same.

7. Conclusions

We have proposed in this paper the AVSE algorithm that divides ants equivalently into different ACO variations. Several simulations were performed optimizing the ball and beam FLC using the proposed AVSE algorithm and each ACO variant individually in which we prove that our proposed method has better performance combining ant partition and a stagnation mechanism. To support the assertions made before, a statistical test was performed. It was found that in this particular FLC for the Ball and Beam system, the optimization of fuzzy rules is not essential, thus it can be overruled. We also compared our algorithm with another related work, concluding that such a comparison favors us even if we used less iterations and ants. One of our contributions is also the methodology and automation for ant partition using a graphic interface. We also reached an informal analysis of an FLC, which can be explored in future works as an independent subject. It is obvious the potential use of this information to determine the performance of a fuzzy logic controller and probably reach a generalization of a problem of optimization, i.e. when it is necessary to know what kind of optimization is needed.

References

[1] C. Blum, Ant colony optimization: introduction and recent trends, Phys. Life Rev. 2 (4) (2005) 353–373.
[2] J. Brownlee, Cleaver Algorithms, Nature-Inspired Programming Recipes, Creative Commons, 2011.
[3] Y.H. Chang, C.W. Chang, H.W. Lin, C.W. Tao, Fuzzy controller design for ball and beam system with an improved ant colony optimization, World Acad. Sci. Eng. Technol. (2009) 616–621.
[4] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evolut. Comput. 1 (1) (2011) 3–18.
[5] A. Djordjevich, in: Grmela, Mastorakis (Eds.), Advances in Intelligent Systems: Fuzzy Systems and Evolutionary Computation, WSEAS Press, USA, 2002.
[6] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization, IEEE Comput. Intell. Mag. (2006) 28–39.
[7] M. Dorigo, L.M. Gambardella, Ant colony system a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evolut. Comput. 1 (1) (1997) 53–66.
[8] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 26 (1) (1996) 29–41.
[9] M. Dorigo, T. Stützle, Ant Colony Optimization, Cambridge Massachusetts Institute of Technology, Bradford, 2004.
[10] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE Service Center, Piscatawa, 1995, pp. 39–43.
[11] A. Engelbrecht, Fundamentals of Computational Swarm Intelligence, Wiley, England, 2005.
[12] M. Galea, Q. Shen, Simultaneous ant colony optimization algorithms for learning linguistic fuzzy rules, in: A. Abraham, C. Grosan, V. Ramos (Eds.), Swarm Intelligence in Data Mining, Springer, Berlin, 2006.

[13] J. Heinonen, F. Pettersson, Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem, Appl. Math. Comput. 187 (2) (2007) 989–998.
[14] J.S.R. Jang, E. Mizutani, C.T. Sun, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, 1997.
[16] H.K. Lam, F.H.F. Leung, P.K.S. Tam, Design of a fuzzy controller for stabilizing a ball-and-beam system, in: Industrial Electronics Society. IECON '99 Proceedings. The 25th Annual Conference of the IEEE, vol. 2, 1999, pp. 520–524.
[17] R. Langari, J. Yen, Fuzzy Logic: Intelligence, Control and Information, Center of Fuzzy Logic, Robotics, and Intelligent Systems, Texas A&M University, Prentice Hall, 1998.
[19] E. Lizárraga, O. Castillo, J. Soria, A Method to Solve the Traveling Salesman Problem Using Ant Colony Optimization Variants with Ant Set Partitioning, in: Recent Advances on Hybrid Intelligent Systems (Studies in Computational Intelligence), Springer, 2013, pp. 237–247.
[20] A.C. Martinez, O. Castillo, O. Montiel, Comparison between ant colony and genetic algorithms for fuzzy system optimization, Soft Comput. Hybrid Intell. Syst. (2008) 71–86.
[21] J. Mendel, Uncertain Rule-Based Fuzzy Logic Systems, Introduction and New Directions, Prentice Hall PTR, 2001.
[22] E. Naredo, O. Castillo, ACO-tuning of a fuzzy controller for the ball and beam problem, in: MICAI, 2011, pp. 58–69.
[23] S.C. Negulescu, C.V. Kifor, Ant colony solving multiple constrains problem: vehicle route allocation, Int. J. Comput. Commun. Control 3 (4) (2008) 366–373.
[24] 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 (Studies in Computational Intelligence), Springer, 2013, pp. 259–273.
[25] W. Pedrycz, Granular Computing: Analysis and Design of Intelligent Systems, CRC Press/Francis Taylor, Boca Raton, 2013.
[26] T. Stützle, S. Linke, Experiments with variants of ant algorithms, Mathware Soft Comput. 9 (2–3) (2002) 193–207.
[29] T. Stützle, H.H. Hoos, MAX–MIN ant system, Future Gener. Comput. Syst. 16 (8) (2000) 889–914.
[31] C.F. Tsai, C.W. Tsai, C.C. Tseng, A new hybrid heuristic approach for solving large traveling salesman problem, Inf. Sci. 166 (1) (2004) 67–81.
[32] L.Y. Tseng, S.C. Liang, A hybrid metaheuristic for the quadratic assignment problem, Comput. Optimiz. Appl. 14 (1) (2006) 85–113.
[34] S.T. Tzen, GA approach for designing fuzzy control with nonlinear ball-and-beam, in: The International Conference on Electrical Engineering, Japan, 2008, pp. 05–09.

Further Reading

[15] C.F. Juang, C. Lo, Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence algorithm, Fuzzy Sets Syst. 159 (21) (2008) 2910–2926.
[18] Y. Li, S. Gong, Dynamic ant colony optimisation for TSP, Int. J. Adv. Manuf. Technol. 22 (2003) 528–533.
[27] T. Stützle, H.H. Hoos, Improving the Ant System: A Detailed Report on the MAXMIN Ant System, Technical Report AIDA-96-12, FG Intellektik, FB Informatik, TU Darmstadt, Germany, 1996.
[28] T. Stützle, H.H. Hoos, MAX–MIN ant system and local search for combinatorial optimization problems, in: S. Voss, S. Martello, I. Osman, C. Roucairol (Eds.), Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publishers, Dordrecht, 1999, pp. 137–154.
[30] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1) (1985) 116–132.
[33] S. Tsutsui, Solving the quadratic assignment problems using parallel ACO with symmetric multi processing, Trans. Jpn. Soc. Artif. Intell. 24 (1) (2009) 46–57.
[35] J. Yen, R. Langari, Fuzzy Logic: Intelligence, Control and Information, Prentice Hall, 2003.
[36] L.A. Zadeh, Fuzzy Logic, IEEE Computer, 1965. pp. 338–353.