Applied Soft Computing Journal 89 (2020) 106095

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Learning rules for Sugeno ANFIS with parametric conjunction operations

Prometeo Cortés-Antonio [a,*], Ildar Batyrshin [b], Alfonso Martínez-Cruz [c], Luis A. Villa-Vargas [b], Marco A. Ramírez-Salinas [b], Imre Rudas [d], Oscar Castillo [a], Herón Molina-Lozano [b]

[a] Tijuana Institute of Technology, Tijuana, Mexico
[b] Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal S/N, Nueva Industrial Vallejo, 07738, Ciudad de México, Mexico
[c] Departamento en Ciencias de la Computación INAOE, Mexico
[d] Obuda University, Budapest, Bécsiút 96b, 1034, Hungary

ARTICLE INFO

Article history:
Received 4 June 2018
Received in revised form 9 January 2020
Accepted 14 January 2020
Available online 20 January 2020

Keywords:
ANFIS
Fuzzy system
Differential evolution algorithm
Learning rule
Parametric fuzzy conjunction
t-norm

ABSTRACT

The paper presents a Sugeno Adaptive Neuro-Fuzzy Inference System with parametric conjunction operations architecture, ANFIS-CX. The advantages of using parametric conjunction operations in fuzzy models are discussed, and learning rules for system identification with such operations are proposed. These learning strategies can include steepest descent gradient, differential evolution and least square estimation algorithms for tuning antecedent, conjunction, and consequent parameters, respectively. The results of system identification by parameter tuning of conjunction operations in addition to or instead of parameter tuning of the input membership functions are presented. Simulation results show that parameter training in conjunction operations, composed of four basic t-norms, significantly improves the approximation capability of fuzzy models.

1. Introduction

Neuro-fuzzy inference systems have been used in a wide range of applications from electro-domestic products such as microwaves, clean machines, among others, to specific applications in medicine, energy, engineering, and economics problems [1–5]. Sugeno models based on an Adaptive Neuro-Fuzzy Inference System (ANFIS) are one of the most popular for such applications [6–14]. This model uses two sources of knowledge. First, it uses a set of fuzzy linguistic rules for representation and processing the expert knowledge describing the behavior of some target system. Second, it uses the knowledge extracted from data through the adaptive neural network design of the fuzzy inference system. A problem with the ANFIS model is that both knowledge and interpretability contained in fuzzy sets often can be lost during the learning process when the adaptive parameters of fuzzy sets changed during optimization of the fuzzy model. The loss of knowledge of fuzzy models has been highlighted in several papers [7,15,16]. In [7], it was mentioned that when there is little data available the possibility of losing expert knowledge is greater if the parameters of membership functions are tuned. On the other hand, in many application works, parameters of membership functions are specified in the structure identification stage by the application of some computational intelligence or data mining techniques, such as clustering, granular computing, fuzzy rule base reduction, classification, and regression trees, amount others [17,18]. In such cases, it cannot desirable to tune the membership function parameters with ANFIS.

An alternative approach to improve the approximation capability of fuzzy models is adding parametric fuzzy operations. Besides, a variety of methodologies to tune these ones can be proposed. Such tuning can be done without or additionally to the tuning of membership functions. Tuning of parametric conjunction and disjunction operations are presented in [15,16,19].

In [19,20], most of the known parametric families of t-norms and t-conorms used for implementing fuzzy operations are defined. Generally, they use complex mathematical operations with high computational cost. These kinds of operations are not desirable in some fuzzy model applications, such as hardware implementations on embedded systems. Alternatively, simpler parametric conjunction operations with low computational cost can

* Corresponding author.
E-mail addresses: prometeo.cortes@tectijuana.edu.mx (P. Cortés-Antonio), batyr1@cic.ipn.mx (I. Batyrshin), amartinezc@inaoep.mx (A. Martínez-Cruz), lvilla@cic.ipn.mx (L.A. Villa-Vargas), mars@cic.ipn.mx (M.A. Ramírez-Salinas), rudas@uni-obuda.hu (I. Rudas), ocastillo@tectijuana.mx (O. Castillo), hmolina@cic.ipn.mx (H. Molina-Lozano).

https://doi.org/10.1016/j.asoc.2020.106095
1568-4946/

be used. In [15,16,21,22], simple conjunction operation families that do not satisfy associativity and/or commutativity properties were proposed and studied, and it was pointed out that in some cases they can be useful, like the following [23]:

- in fuzzy inference systems with two inputs and one output;
- when the positions of input variables in fuzzy rules are fixed in the inference process;
- in problems where the non-commutativity property is desirable to reflect different influences of the input variables on the output of the system;
- in systems where the equivalent transformations of logical forms are not needed, such as, in most of the applications found in the literature.

With the main goal of getting more capability to approximate the target system, fuzzy models with parametric fuzzy conjunctions have been implemented in both software and hardware realizations [15,19,23–33]. A Mamdani Fuzzy Model with parametric conjunctions was implemented without machine learning techniques in [24–26]. In [24], the influence of tuning the conjunction operations parameters versus membership function parameters is compared on the stability and robustness of fuzzy control systems. In [25], the conjunction operation parameters together with type-2 fuzzy sets parameters were tuned by trial and error method in the optimization of Mamdani fuzzy models. In [27], a Mamdani model with parametric conjunctions was implemented, and a learning rule was proposed to improve the accuracy of the model, and by the use of three prediction applications, it was tested to study the rules cooperation using an evolvable optimization technique. In [28], a neuro-fuzzy Mamdani model was studied in a classification problem by using a gradient method for the optimization of parametric conjunction operations. Sugeno fuzzy models with parametric conjunction operations have been implemented in both modeling [15] and control applications [29–31], where the accuracy improvement of the model was achieved by tuning all model parameters with a gradient method. Other kinds of neuro-fuzzy models with parametric conjunction operations were optimized by memetic algorithms in [32,33].

Note, that there exist other parametric functions to implement fuzzy operations used in fuzzy modeling, such as copulas [34,35] and uninorms [36].

Most of the mentioned works on fuzzy models with parametric fuzzy operations show that these models have better approximation capabilities versus models that no include them. However, these works do not take the advantages of the mentioned ANFIS models. And so, it not is clear, if the approximation capability of traditional ANFIS is better than fuzzy models with parametric fuzzy operations.

The main motivations of this paper are, on the one hand, to introduce two fuzzy models that contain parametric fuzzy operations including conjunction operations adequate for hardware implementation [37]. On the other hand, a learning rule bases on ANFIS with different levels of constrains is developed, which is highlighted by a separation of linear and nonlinear parameters, in order to the model can converge faster in the training phase. Moreover, we develop an analysis to measure the improvement of the ANFIS with parametric conjunction operations compared with ANFIS models that do not include them.

Thus, we propose a hybrid learning rule for parameter identification, which uses steepest descent gradient and least square estimation methods for tuning antecedent and consequent parameters respectively, and additionally, the differential evolution algorithm (DEA) is used for the identification of conjunction operation parameters to obtain models with greater approximation capability. We have decided to choose DEA because of the parametric conjunction operations used in our model are not continuous functions. Also, DEA is simple and only uses arithmetic operations, and in some works, it was proved that DEA has better performance in many searching problems than other non-gradient optimization heuristics [38–41].

The paper has the following structure. The basics of fuzzy conjunction operations and t-norms used to generate fuzzy parametric conjunctions are presented in Section 2. Section 3 discusses soft computing techniques, such as Sugeno and ANFIS models and the differential evolution algorithm. Section 4 introduces two variants of the ANFIS model, called ANFIS-CX, and three choices of learning rules. In Section 5, we study the sensitivity regions, where conjunction parameters influence the output of the fuzzy system. The results of the study ANFIS-CX and its learning rules in two numerical examples are given in Section 6. The final section contains conclusions and a discussion of future works.

2. Fuzzy conjunctions and t-norms

2.1. Basic definitions

Consider the function T:[0,1]×[0,1]→[0,1] with the following possible properties on [0,1]:

$$P_1.T(a,1)=a,\ \ T(1,b)=b, \qquad \text{(boundary conditions)}$$

$$P_2.T(a,b)\le T(c,d),\ \ \textit{if}\ a\le c,\ b\le d \qquad \text{(monotonicity)}$$

$$P_3.T(a,b)=T(b,a) \qquad \text{(commutativity)}$$

$$P_4.T(a,T(b,c))=T(T(a,b),c) \qquad \text{(associativity)}$$

$$P_5.T(a,b)\le \min(a,b) \qquad \text{(range condition)}$$

This function will be called a conjunction, a commutative conjunction, a t-norm or a t-subnorm [19,42] if the following properties are fulfilled for all a, b, c ∈ [0,1], correspondingly: $\{P_1,P_2\}$, $\{P_1–P_3\}$, $\{P_1–P_4\}$, $\{P_2–P_5\}$. For any conjunction from P_1, P_2, it follows for all a,b ∈ [0,1]:

$$T(a,0)=0,\quad T(0,b)=0$$

Below the simplest t-norms that will be considered as basic t-norms in the generation of fuzzy parametric conjunctions and t-norms are presented:

$$T_D(a,b)=\begin{cases}a, & \textit{if}\quad b=1\\ b, & \textit{if}\quad a=1\\ 0, & \textit{if}\quad a,b<1\end{cases} \qquad \text{(drastic product)} \quad (1a)$$

$$T_L(a,b)=\max(a+b-1,0) \qquad \text{(Lukasiewicz t-norm)} \quad (1b)$$

$$T_P(a,b)=ab \qquad \text{(product)} \quad (1c)$$

$$T_M(a,b)=\min(a,b) \qquad \text{(minimum)} \quad (1d)$$

These t-norms have efficient hardware implementation [22,25,37,43] because they use in their definitions very simple mathematical operations. Fig. 1 depicts the shapes of these t-norms defined on [0, 1].

Denote $T_1\le T_2$, *if* $T_1(a,b)\le T_2(a,b)$ for all a, b in [0, 1]. For any conjunction T and for the above basic t-norms we have:

$$T_D\le T\le T_M,\qquad T_D\le T_L\le T_P\le T_M, \qquad (2)$$

From $T\le T_M$ it follows that any t-norm is a t-subnorm.

2.2. Fuzzy parametric conjunctions based on monotone sum of t-norms

Let $L=[0,1]$ and $p\in[0,1)$ is a parameter, then p divides L on 2 intervals: $X_1=[0,p]$, $X_2=(p,1]$, and all domain $L\times L$ on 4 sectors: $D_1=X_1{\times}X_1$, $D_2=X_2\times X_1$, $D_3=X_1\times X_2$, $D_4=X_2\times X_2$, see Fig. 2a. When $p=0$ we only have one interval $X_2=[0,1]$

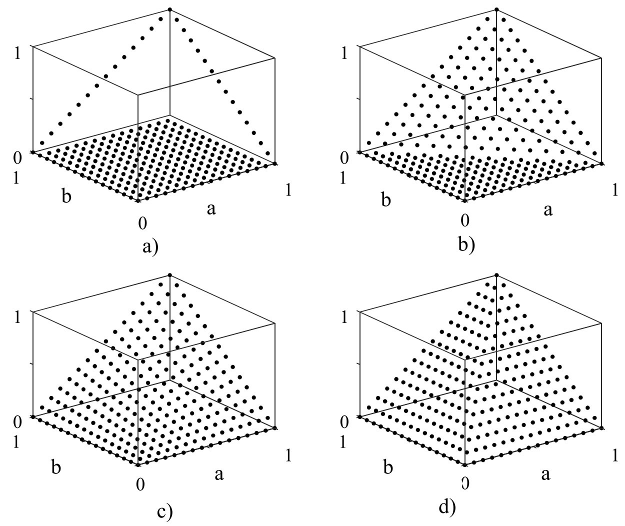

Fig. 1. Simplest t-norms: (a) Drastic; (b) Lukasiewicz; (c) Product; (d) Minimum.

and one sector $D_4 = X_2 \times X_2 = L \times L$ coinciding with all domain $L \times L$. In this case all other sectors D_1, D_2, D_3 will be empty.

Below we consider a method for constructing parametric conjunction operations T on L by means of basic t-norms T_i defined on sectors D_i, $i \in \{1, 2, 3, 4\}$ [37].

Method 4T. This method can use two, three or four different t-norms. Let T_i, $i \in \{1, 2, 3, 4\}$, are t-norms defined on $L = [0, 1]$ such that:

$$T_1 \leq T_2 \leq T_4, \quad T_1 \leq T_3 \leq T_4. \tag{3}$$

Then the function T: $L \times LL$ defined by:

$$T(a, b; p) = \begin{cases} T_1(a, b), & \text{if} \quad a, b \leq p \\ T_2(a, b), & \text{if} \quad p < a, \text{ and } b \leq p \\ T_3(a, b), & \text{if} \quad a \leq p, \text{ and } p < b \\ T_4(a, b), & \text{if} \quad p < a, b \end{cases} \tag{4}$$

is a conjunction, where $p \in [0,1)$ is a parameter. Fig. 2b depicts the location of basic t-norms used in this method in segments of $L \times L$. The conjunction $T(a, b; p)$ is commutative if all T_i are commutative and $T_2 = T_3$. Note that when $p = 0$ the conjunction T equals to T_4 defined on all the input domain $L \times L$. The method 4T of generation of conjunctions is based on the p-monotone sum of t-norms [44]. Table 1 shows all possible commutative parametric conjunction operations generated by the 4T method from the combination of Drastic (D), Lukasiewicz (L), Product (P), Minimum (M) basic t-norms. The improvement in the approximation

Table 1
Commutative parametric conjunction operations generated by method 4T.

Number	Type	Number	Type	Number	Type	Number	Type
1	DDDL	5	DLLP	9	DMMM	13	LPPM
2	DDDP	6	DLLM	10	LLLP	14	LMMM
3	DDDM	7	DPPP	11	LLLM	15	PPPM
4	DLLL	8	DPPM	12	LPPP	16	PMMM

capability of fuzzy models that add the conjunction operation as shown in Table 1 will be analyzed in Section 6.

Note that the parametric conjunction operations can be applied to a fuzzy system with more than two inputs by successively applying the operations. However, it should be considered that the operations presented in Table 1 do not satisfy the associativity property.

3. Soft computing techniques

3.1. Sugeno Fuzzy inference system

In this paper, we use the first-order Sugeno fuzzy inference system [7,45]. For a Sugeno model with n-inputs, the fuzzy rule base can have the following structure:

$$R^i : \textit{if } x_1 \textit{ is } X_1^i \textit{ and } x_2 \textit{ is } X_2^i \ldots \textit{ and } x_n \textit{ is } X_n^i \textit{ then}$$
$$y^i = \theta_1^i x_1 + \theta_2^i x_2 + \cdots \theta_n^i x_n + \theta_{n+1}^i, \quad (i = 1, \ldots, m) \tag{5}$$

where $x_1, x_2, \ldots, x_n$ are input variables, and $X_1^i \in \{X_1^1, \ldots, X_1^{m_1}\}$, $X_2^i \in \{X_2^1, \ldots, X_2^{m_2}\}, \ldots, X_n^i \in \{X_n^1, \ldots, X_n^{m_n}\}$ are linguistic (fuzzy) values, and $m_1, m_2, \ldots, m_n$ are the number of fuzzy values of each input variable, and m is the number of fuzzy rules.

Generally, a fuzzy value is defined by a parameterized membership function, and these parameters are called as the antecedent parameters of a fuzzy system. Some membership functions useful to define fuzzy values can be triangular, trapezoidal, gbell, among others [7]. As an example, a three-parameter triangular membership function to define the fuzzy value i of input x_1 can be denoted as follows (with $\alpha_{1,1}^i < \alpha_{1,2}^i < \alpha_{1,3}^i$):

$$\mu_{X_1^i}(x_1) = \mu_{X_1^i}\left(x_1; \alpha_{1,1}^i, \alpha_{1,2}^i, \alpha_{1,3}^i\right)$$
$$= \begin{cases} 0 & x_1 \leq \alpha_{1,1}^i \\ \dfrac{x_1 - \alpha_{1,1}^i}{\alpha_{1,2}^i - \alpha_{1,1}^i} & \alpha_{1,1}^i \leq x_1 \leq \alpha_{1,2}^i \\ \dfrac{\alpha_{1,3}^i - x_1}{\alpha_{1,3}^i - \alpha_{1,2}^i} & \alpha_{1,2}^i \leq x_1 \leq \alpha_{1,3}^i \\ 0 & \alpha_{1,3}^i \leq x_1 \end{cases} \tag{6}$$

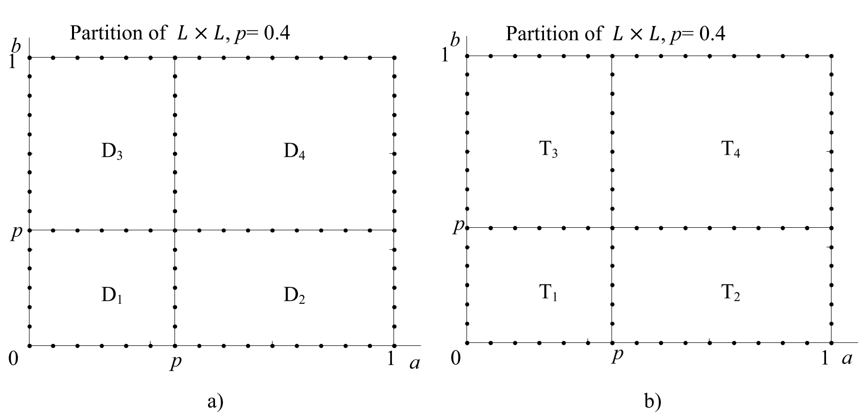

Fig. 2. (a) Partition of $L \times L$ on segments defined by parameter p. (b) Method 4T of constructing conjunctions.

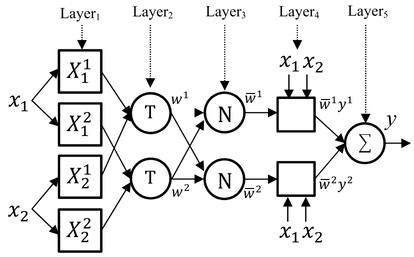

Fig. 3. Traditional ANFIS model with adaptive nodes in layers 1 and 4.

The set of antecedent parameters into a fuzzy rule base can be denoted as a vector $\alpha = [\alpha_{1,1}^1, \ldots, \alpha_{1,p11}^1, \alpha_{1,1}^2, \ldots, \alpha_{1,p12}^2, \ldots, \alpha_{1,1}^{m_1}, \ldots, \alpha_{1,p1m_1}^{m_1}, \quad \ldots, \alpha_{n,pnm_n}^{m_n}]$. Where the first subscript indicates the parameter that belongs to a fuzzy input variable j, the superscript indicates its fuzzy value and the second subscript helps to identify the parameter number.

The firing strength of rule R^i, called w^i, indicates the degree to which the antecedent part of a rule is satisfied. w^i equals to the conjunction of the membership values of the input fuzzy sets X_j^i is defined as follows:

$$w^i = \mathop{\mathbf{T}}_{j=1}^{n} \mu_{X_j^i}(x_j) \tag{7}$$

The consequent part of the rule, y^i, is a crisp value defined by a parameterized linear function. These parameters are called as the consequent parameters of the fuzzy system. In this work, these parameters are denoted as θ_j^i, and the set of all consequent parameters is denoted as θ.

The overall output y of a fuzzy system with m rules can be computed as follows:

$$y = \frac{\sum_{i=1}^{m} w^i y^i}{\sum_{i=1}^{m} w^i} \tag{8}$$

The major advantage of this model is that the defuzzification stage is not required as in a Mamdani model, and the optimal values of the set of consequent parameters, θ, can be identified by some linear optimization method. Therefore, this model is more suitable for applications when a data set is available.

If the firing strength w^i is normalized as follows:

$$\overline{w}^i = \frac{w^i}{\sum_{j=1}^{m} w^j} \tag{9}$$

The output equation of a Sugeno fuzzy system can be rewritten in terms of normalized firing strengths as follows:

$$y = \sum_{i=1}^{m} \overline{w}^i y^i \tag{10}$$

3.2. ANFIS Model

In general, the ANFIS model is any adaptive neuro-fuzzy inference system implemented in the framework of artificial neural networks. In particular, Sugeno-ANFIS is a Sugeno fuzzy inference system implemented into an adaptive feedforward neural network architecture with five layers to perform fuzzy operations. For simplicity in Fig. 3, it is shown a traditional ANFIS architecture of 2-inputs and 1-output with 2 rules. Layer_1 and Layer_4 contain the adaptive nodes (represented by squares) where the antecedent and consequent parameters can be tuned. The circles denote the fixed nodes. Here, T refers to product t-norm, N to normalization of firing strength, and Σ to the summation. The generalization to multiple inputs and multiple rules is direct.

The output of the ANFIS model of two-input and one-output and two rules can be expressed as a linear combination of consequent parameters as follows:

$$\begin{aligned} y &= \overline{w}^1\left[\theta_1^1 x_1 + \theta_2^1 x_2 + \theta_3^1\right] + \overline{w}^2\left[\theta_1^2 x_1 + \theta_2^2 x_2 + \theta_3^2\right] \\ y &= \overline{w}^1 x_1 \theta_1^1 + \overline{w}^1 x_2 \theta_2^1 + \overline{w}^1 \theta_3^1 + \overline{w}^2 x_1 \theta_1^2 + \overline{w}^2 x_2 \theta_2^2 + \overline{w}^2 \theta_3^2 \end{aligned} \tag{11}$$

The overall output of ANFIS for a system with n-inputs and m-rules can be generalized as Eq. (12):

$$\begin{aligned} y = {} & \overline{w}^1\left[\theta_1^1 x_1 + \theta_2^1 x_2 + \cdots + \theta_n^1 x_n + \theta_{n+1}^1\right] + \cdots + \\ & \overline{w}^m\left[\theta_1^m x_1 + \theta_2^m x_2 + \cdots + \theta_n^m x_n + \theta_{n+1}^m\right] \\ y = {} & \overline{w}^1 x_1 \theta_1^1 + \overline{w}^1 x_2 \theta_2^1 + \cdots + \overline{w}^1 x_n \theta_n^1 + \overline{w}^1 \theta_{n+1}^1 + \cdots + \\ & \overline{w}^m x_1 \theta_1^m + \overline{w}^m x_2 \theta_2^m + \cdots + \overline{w}^m x_n \theta_n^m + \overline{w}^m \theta_{n+1}^m \end{aligned} \tag{12}$$

To identify the parameters of ANFIS by using a training set obtained from a target system, in [6], an iterative hybrid learning rule was proposed to train separately the antecedent and consequent parameters by a combination of two trained stages: *i*) gradient descent (GD) and; *ii*) least square estimation (LSE) methods respectively.

In the first stage, the set of consequent parameters θ, are tuned by some LSE method, and the set of antecedent parameters α is considered fixed. In the second stage, the parameters of α are tuned by some GD method, whereas the parameters of θ are considered fixed.

Given a training data set with Q data pairs $\left\{\left(\boldsymbol{x_q} \vdots \boldsymbol{t_q}\right),\quad q = 1, 2, \ldots, Q\right\}$, and by substituting each data pair into Eq. (12), it yields to a set of Q linear equations as follows:

$$\begin{aligned} t_1 &= \overline{w}^1 x_{1,1} \theta_1^1 + \overline{w}^1 x_{2,1} \theta_2^1 + \cdots + \overline{w}^m x_{n,1} \theta_n^m + \overline{w}^m \theta_{n+1}^m \\ t_2 &= \overline{w}^1 x_{1,2} \theta_1^1 + \overline{w}^1 x_{1,2} \theta_2^1 + \cdots + \overline{w}^m x_{n,2} \theta_n^m + \overline{w}^m \theta_{n+1}^m \\ &\ \ \vdots \\ t_Q &= \overline{w}^1 x_{1,Q} \theta_1^1 + \overline{w}^1 x_{2,Q} \theta_2^1 + \cdots + \overline{w}^m x_{n,Q} \theta_n^m + \overline{w}^m \theta_{n+1}^m \end{aligned} \tag{13}$$

In matrix notation, we can rewrite the system of linear equations given at (13) as follows:

$$t = A\theta \tag{14}$$

Where the set of unknown linear consequent parameter θ, in vector representation, is expressed as follows:

$$\theta = [\theta_1^1, \theta_2^1, \ldots, \theta_n^1, \theta_{n+1}^1, \ldots, \theta_1^m, \theta_2^m, \ldots, \theta_n^m, \theta_{n+1}^m]^T$$

the size of θ is $k = m(n+1)$. And the design matrix A is a $Q * k$ matrix:

$$A = \begin{bmatrix} \overline{w}^1 x_{1,1} & \cdots & \overline{w}^m \\ \vdots & \ddots & \vdots \\ \overline{w}^1 x_{1,Q} & \cdots & \overline{w}^m \end{bmatrix}$$

The q-row of the joint data matrix $\left[A \vdots t\right]$, denoted by $\left[{a_q}^T \vdots t_q\right]$, is related to the q *th* input–output data pair $\left(\boldsymbol{x_q} \vdots t_q\right)$. The recursive LSE is an efficient method to reduce the computation on the tuning of the consequent parameters, and it uses the next pair of iterative equations [6,7]:

$$P_{q+1} = P_q - \frac{P_q a_{q+1} {a_{q+1}}^T P_q}{1 + {a_{q+1}}^T P_q a_{q+1}} \tag{15a}$$

$$\theta_{q+1} = \theta_q + P_{q+1} a_{q+1}\left(t_q - {a_{q+1}}^T \theta_q\right) \tag{15b}$$

where $q = 1, 2, \ldots, Q$, and the vector θ_q contains the values of the k consequent parameters at the q-iteration, and the estimation gain matrix, P, is a matrix of $k \times k$ dimension. The initial values of the iterative equations are $\theta_0 = 0$, $P_0 = \lambda I$, where λ is a positive large number and I is the identity matrix. The optimal values of consequent parameters are given at Q-iteration, that is when $\theta^* = \theta_Q$. On the other hand, the steepest gradient descend and the chain rule learning is used for tuning the set of antecedent parameters, α. First, a gradient vector is obtained recursively by applying the chain rule of the error measure concerning each parameter $\alpha^i_{j,k}$. The error measure used in this work is as follows:

$$E_q = (t_q - y_q)^2 \tag{16}$$

where t_q and y_q are the q-target output and the actual output produced by the presentation of the q-input vector $\boldsymbol{x}_q$. The actual output of the ANFIS can be expressed as a function of the antecedent parameters as follows:

$$y = \frac{\sum_{i=1}^{m} y^i \left[\underset{j=1}{\overset{n}{\mathbf{T}}} \mu_{x_j^i}(x_j) \right]}{\sum_{i=1}^{m} \left[\underset{j=1}{\overset{n}{\mathbf{T}}} \mu_{x_j^i}(x_j) \right]} = \frac{\sum_{i=1}^{m} y^i \left[\underset{j=1}{\overset{n}{\mathbf{T}}} \mu_{x_j^i}(x_j; \alpha^i_{j,2}, \dots, \alpha^i_{j,pji}) \right]}{\sum_{i=1}^{m} \left[\underset{j=1}{\overset{n}{\mathbf{T}}} \mu_{x_j^i}(x_j; \alpha^i_{j,2}, \dots, \alpha^i_{j,pji}) \right]} \tag{17}$$

Hence the overall error when all the data set is considered is given as follows:

$$E = \sum_{q=1}^{Q} E_q \tag{18}$$

Then, the derivative for each antecedent parameter $\alpha^i_{j,k}$ of the overall error measure E_j, which corresponds to an item of gradient vector is:

$$\frac{\partial E_j}{\partial \alpha^i_{j,k}} = \frac{\partial \sum_{q=1}^{Q} (t_q - y_q)^2}{\partial \alpha^i_{j,k}} \tag{19}$$

The update formula for a generic parameter $\alpha^i_{j,k}$ is:

$$\Delta \alpha^i_{j,k} = -\eta \frac{\partial E}{\partial \alpha^i_{j,k}} \tag{20}$$

and η is a positive number defining the rate change of parameter $\alpha^i_{j,k}$ [6,7]. The separation of nonlinear and linear parameters in the ANFIS hybrid learning rule causes its effective approximation capability and efficient convergence. To the best of our knowledge the ANFIS hybrid learning rule [6,7] has not been outperformed in other works and nowadays it is used in hundreds of applications, see, e.g. [46–49]. For this reason, in this paper, we use the ANFIS hybrid learning rule as a baseline for evaluation of the performance of our method.

3.3. Differential evolution algorithm

In complex mathematics problems and real-world engineering applications, the search for an optimal solution using traditional optimization methods, for example, gradient-based optimization techniques, or an exhaustive search, generally is not possible or impractical. Optimization algorithms based on metaheuristics such as Genetic Algorithms, Distribution Estimation Algorithms, Differential Evolution Algorithms, Particle Swarm Optimization, Ant Colony Optimization, among others, have been widely accepted and used in the optimization of real problems in engineering, economics, business applications, among others [50–53].

The Differential Evolution Algorithm (DEA) was introduced as a simple population-based algorithm [38,54]. It uses only three control parameters and performs only basic operations such as addition, subtraction, multiplication, and comparison. The DEA's performance is competitive and more efficient compared with other evolutionary algorithms or optimization metaheuristics in many problems for real representation variables [39,55,56].

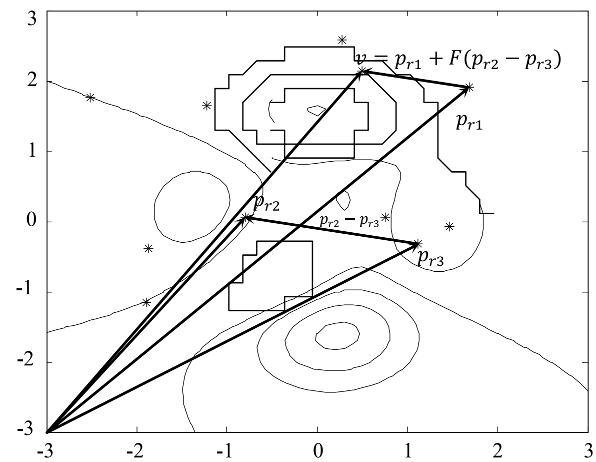

Fig. 4. Generation of a mutated vector of a two-dimension case.

In general, the DEA's goal is to find a global optimal solution of a function in a continuous space. In particular, and without loss of generality, this problem can be reduced to the problem of finding the global minimum of a function. The simple DEA, proposed in [54,57], used three configuration parameters: (1) a crossover operator *CR*; (2) an escalation factor *F*; and (3) a population size *NP*.

Generally, to start the simple DEA, the configuration parameters are set up, an initial population is randomly generated, and the fitness of each solution and best individual are calculated using a predefined fitness function $f(.)$.

The interactive process of mutation, crossover and selection operations in the simple DEA is as follows: for each individual p_i from the current population P, a mutated vector v is generated by three mutually different solutions and different to individual p_i (see line 7, Algorithm 1). Then the crossover operation mixes the mutated vector with p_i, to generate a trial individual v, as it is shown through lines 9 to 14. If the fitness of the trial solution $f(v)$, is better than $f(p_i)$, then the trial vector replaces the solution p_i in the current population, otherwise, p_i still belongs to the population. This process will be repeated for all individuals of the current population. The algorithm finishes until a stopping criterion is satisfied.

Algorithm 1 shows the pseudocode to implement the DEA and Fig. 4 illustrates a two-dimensional case with a population of 10 solutions or individuals. The algorithm selects p_{r2} and p_{r3} individuals and compute the difference between them. After that, the mutated vector v is the sum of the value of p_{r1} and the scaled difference of the p_{r2} and p_{r3} individuals. Through a fitness function will be determined if v is or is not a better solution than p_i.

The DEA uses the same stopping conditions as other optimization algorithms: the maximum number of generations; an algorithmic time; the difference between sequential errors; and/or a minimum error.

In this paper, the DEA will be used for finding the optimal conjunction operation parameters implemented into ANFIS architecture, where the cost function is based on the root mean square error between the target system and the model. In the next section, three different representations of conjunction parameters are explained. Other approaches that perform some synergistic or hybridization soft computing techniques to add more intelligent behavior to the model have been proposed in different applications and optimization problems [58–62].

Set up the control parameters and stop condition values: NP, Cr, F *and* $Gmax, ErrorMin$
Create randomly an initial population P *of size* NP, each individual has D-dimensionality
Deternine and compute the fitness $f(.)$ *for each individual of the population, and*
find the individual with best fitness
while *stop condition is not satisfied*
    for *each individual* p_i
        select randomly three individuals from P: $p_{r_1} \neq p_{r_2} \neq p_{r_3} \neq p_i$
        generate a mutated vector: $v = p_{r_1} + F\,(p_{r_2} - p_{r_3})$
        generate a default element: $jrand = randInt[1{:}D]$
        for *each dimension of the test vector* ***do***
            if $(rand[0,1] > CR$ ***or*** $dimension \neq jrand)$ ***then***
                $v_d = p_{i,d}$
            end if
        end for
        if $\left(f(v) < f(p_i)\right)$ ***then***
        Replace $p_i, f(p_i)$ by v and $f(v)$ respectively
    end for
end while

Algorithm 1: Differential Evolution algorithm pseudocode

4. ANFIS-CX: The architectures and learning rules

In this paper, two variants of the ANFIS architecture are implemented. The first one uses adaptive nodes only in layer_2 and layer_4 to specify adaptive functions with conjunction and consequent parameters, respectively (see Fig. 5a). It is motivated by the cases when we would like to retain the knowledge base embedded into the input fuzzy sets or when we are working with fuzzy sets that have been already trained in a previous stage of the design. The second ANFIS architecture is useful when input sets are defined by applying simple common-sense rules or when the training data set is large, or defined by trial and error. Therefore, the ANFIS architecture can have adaptive nodes in layer_1 additionally to the adaptive nodes considered in the first model. Fig. 5 shows these architectures; they will be referred to as ANFIS-C1 and ANFIS-C2 respectively (The letter C refers that ANFIS models include parametric conjunction operations).

The ANFIS-CX architectures studied in this paper have the possibility of using: (a) 16 different parametric conjunction operations at layer_2, generated by the method 4T described in Section 2 (see Table 1); (b) triangular or generalized bell membership functions in layer_1; (c) parametric linear function in layer_4; (d) three different number of granules for each input variable to create the rule base using grid partition.

We are going to introduce three different learning rules for processing fuzzy systems, which are based according to the constraints imposed on conjunction operations used in fuzzy rules. The first version uses a unique parameter value $p \in [0, 1)$ for the unique parametric conjunction operation for all rules of the fuzzy model. The second one also uses the same parametric conjunction operation for all rules, but parameter values in p can be different in these rules. In General, there are m different parameter values $p \in [0, 1)$ in m rules. For the last one, the model can have different parametric conjunction operations with independent parameter values in the rule base. Therefore, each rule contains two conjunction parameters defining: (1) the type of conjunction operation from method 4T (see Section 2.2), and (2) the value $p \in [0, 1)$.

The representation of the solutions in DEA, also called chromosome, in each version is as follows. In version 1, an individual p_i contains only one real value $p \in [0, 1)$. In version 2, $\boldsymbol{p_i}$ is represented by a vector $\boldsymbol{p_i} = [p_{i,1}, \ldots, p_{i,m}]$, where $p_{i,l}$ determines the parameter $p \in [0, 1)$ in the lth rule. In version 3, each individual $\boldsymbol{p_i}$ of the population is represented by a vector of m elements and each element is composed by a pair of parameters indicating: (1) the operator type with the discrete value $\tau \in \{0, \ldots, 15\}$, and (2) the value $p \in [0, 1)$. Table 2 summarizes these three different ways to use the learning process.

The justification for using different conjunction operators in a fuzzy model is to allow that a rule, instruction or sentences into the model can be processed in different ways, similar to as different humans can perform correctly a same instruction or task in different ways.

A fuzzy model obtained by the learning rule in version 1 can be interpreted as a rigid system described by rules processed in the same way. A fuzzy model obtained by learning rules with versions 2 or 3, shows flexible behavior, and it can perform each task differently. Therefore, the type of uncertainty quantification carried out in parametric conjunction operations face to the ambiguity and non-specificity embedded in the way to process a rule, which is different to the uncertainty studied in others approaches, such as type-2 fuzzy sets or Intuitionistic fuzzy sets [63–65]. This uncertainty is similar to that presented when people are solving a complex problem, a task can be resolved in different ways.

We will explain the learning processes for training the parameters of the ANFIS-C1 model, and after that, we describe the learning rules for training the ANFIS-C2, which uses the traditional hybrid ANFIS and ANFIS-C1 learning rules in two stages for tuning all its parameters.

A learning rule for ANFIS-C1 models uses DEA and LSE optimization methods. LSE is used in the calculation process of the fitness function of the DEA. At each iteration (or generation) and for each individual of the population, DEA generates a trial solution vector v, according to both the learning rule version and the algorithm described in Section 3.3. Then, to compute the fitness function $f(v)$, v is considered as a fixed value in the fitness function, and $f(v)$ can be got by the following two steps:

(a) Find the optimal values of the consequent parameters θ^*, using the trial solution v as the conjunction parameters values and training data set, by a least square estimator method.
(b) Compute $f(v)$ equal to the root mean square error between the output y of the model and the output t of the trained data set using batch mode.

After that, DEA runs the selection operation and repeats the iteration process. Note, that when $f(v)$ has better fitness than $f(p_i)$, not only conjunction parameters are updated, but also consequent parameters.

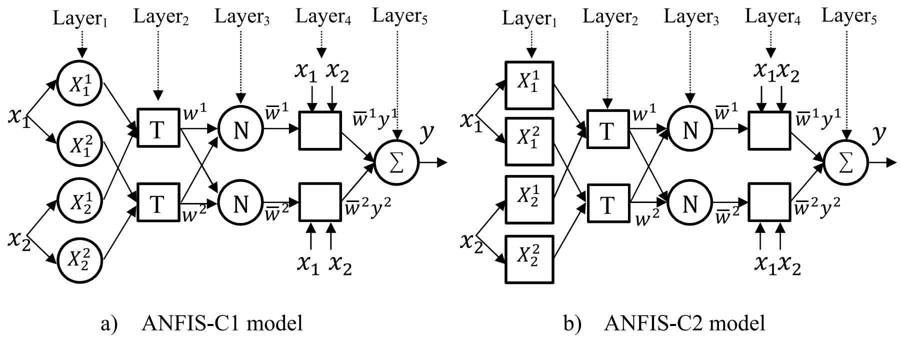

Fig. 5. Architecture of Sugeno ANFIS with parametric conjunction operations.

Table 2
Representation of individuals in learning rule versions.

Version	Version description	Number of conjunction parameters	Representation of an individual
ANFIS-CX-V1	One parametric conjunction operation and one parameter value for all rule base	1	$p_i \in [0, 1)$
ANFIS-CX-V2	One parametric conjunction operation and different parameter values in m rules	m	$\boldsymbol{p_i} = [p_{i,1}, \ldots, p_{i,m}]$, $\boldsymbol{p_i} \in [0, 1)^m$
ANFIS-CX-V3	Different parametric conjunction operations and different parameter values in m rules	$2m$	$\boldsymbol{p_i} = [(\tau_{i,1}, p_{i,1}), \ldots, (\tau_{i,n}, p_{i,m})]$ $\tau_{i,l} \in \{0, \ldots, 15\}$, $p_{i,l} \in [0, 1)$, $l = 1, \ldots, m$

A learning rule for ANFIS-C2 is processed in two stages. In the first one, traditional Sugeno-ANFIS learning rule is applied, remember that it always uses fixed t-norms (product t-norms). Then, the hybrid learning rule identifies antecedent and consequents parameters as in [6]. In the second one, the optimized antecedent parameters, found in stage one, are used as fixed parameters in the model, and the product t-norms are substituted by some parametric conjunction operation. And thereby the major task in the second stage is the tuning of conjunction and consequent parameters. Therefore, in this stage, the learning process is simply to apply some version of the learning rule mentioned to identify an ANFIS-C1.

The learning rules proposed for ANFIS-C2 models are called composed hybrid learning rules and can be summarized by the next steps:

1. Implement and learn a traditional ANFIS model.
2. Choose a learning rule version to determine the constraints for the rule base of the model. If learning rule versions 1 or 2 are chosen, then it should be chosen as a parameter conjunction operation of Table 1, which will be the conjunction operation used for the ANFIS-C2.
3. Tune the conjunction operation and consequent parameters.

The batch mode learning approach is used in the tests presented in Section 6, i.e., the overall training set is used to update the parameters. Note that online learning could be used in other kinds of applications.

5. Sensitivity regions of conjunction operations

In this section, we review the input regions, where the use of different parametric fuzzy conjunction operations can change or modify the output of the fuzzy model. According to Jang in [7], the antecedents of fuzzy rules divide the input space into fuzzy regions. Here, we consider a division of the input space as a local

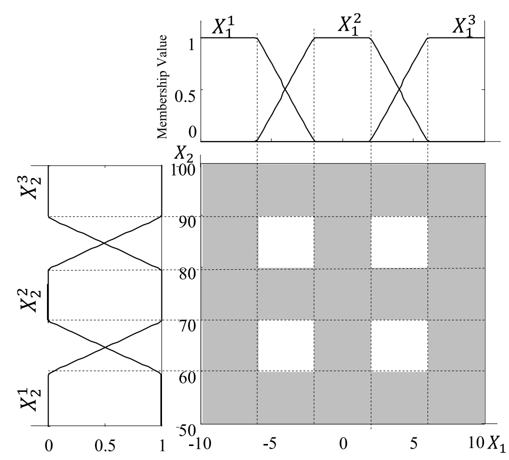

Fig. 6. Sensitivity regions (white color) in the input space of fuzzy system with trapezoidal membership functions.

fuzzy sensitivity region which is defined by each rule of the fuzzy model, and it is dependent not only on the input membership functions but also on parametric fuzzy conjunction operations.

Let the antecedent of a fuzzy rule R^i of a fuzzy system of n-inputs with fuzzy sets $X_1^i, \ldots X_n^i$ be related by fuzzy parametric conjunction operations, the local sensitivity regions, Sen_{R^i}, is defined as follows:

$$Sen_{R^i} = \left\{ (x_1, \ldots, x_n) \mid x_1 \in X_1, \ldots, \; x_n \in X_n, 0 < \mu_{X_j^i}\left(x_j\right) < 1 \right\} \tag{21}$$

for $j = 1, \ldots, n$. The overall output of the fuzzy model can be sensitive to a change of either a fuzzy conjunction operation or its parameters. The union of the sets formed by all rules of the fuzzy system will be called the *sensitivity region of fuzzy conjunctions*.

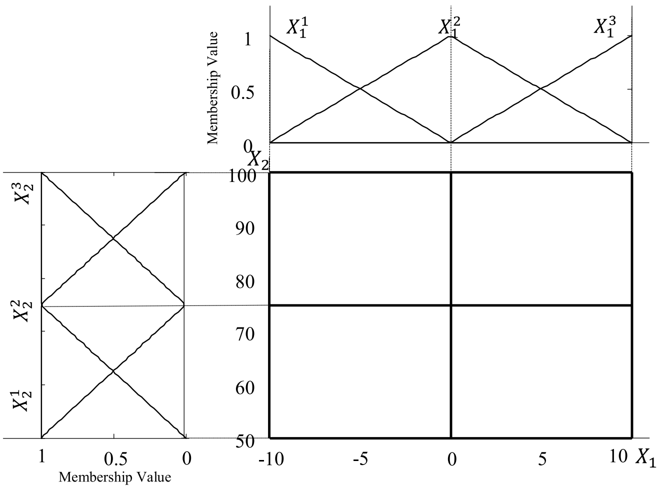

Fig. 7. Sensitivity region in the input space of fuzzy system with triangular membership functions.

The *sensitivity* region is more illustrative for the case of a 2-inputs fuzzy system with fuzzy sets X_1^i and X_2^i, where the local sensitivity region is defined as follows:

$$Sen_{R^i} = \left\{(x_1, x_2) \mid x_1 \in X_1 \text{ and } x_2 \in X_2,\ 0 < \mu_{X_1^i}(x_1)\ ,\ \mu_{X_2^i}(x_1) < 1\right\} \tag{22}$$

Fig. 6 depicts an example of such sensitivity region of fuzzy conjunctions given by four white blocks for an example of a fuzzy system with trapezoidal membership functions.

Fig. 7 shows the sensitivity region of fuzzy conjunctions (with white color) for an example of a fuzzy system with triangular membership functions. As one can note, in comparison with trapezoidal membership functions the triangular membership functions allow to have larger sensitivity region in the input space and hence a tuning of fuzzy parametric conjunctions in the rules of such fuzzy system can have a larger influence on the output of the system increasing its ability in modeling real data. These observations are consistent with the ones in [24]

6. Results

To measure the performance of the proposed models, we use the well-known root mean square error (RMSE) between the desired output target system and the actual model output. Besides, we measure the improvement of the approximation capability of our model, based on the Improvement Percentage Index (IP) [27]:

$$IP = \frac{Initial_{RMSE} - Final_{RMSE}}{Initial_{RMSE}} * 100\% \tag{23}$$

where Initial RMSE and Final RMSE are obtained as follows:

(i) For ANFIS-C1, the Improvement Index IP_1 is calculated as follows. The $Initial_{RMSE1}$ is the error obtained for the initial Sugeno model set up with membership functions that uniformly partitioned the input space, and all conjunction operations set up as product t-norms, and with the consequent parameters optimized by LSE. The $Final_{RMSE1}$ is obtained from the optimized ANFIS-C1. IP_1 indicates the improvement performed by the ANFIS-C1 when compared to the initial Sugeno model (trained by LSE).

(ii) For ANFIS-C2, we calculate two Improvement Percentage Indexes. The Global Improvement Index, IP_G, which uses the $Initial_{RMSE1}$ considered above and the $Final_{RMSE2}$ equal to the RMSE obtained after the fuzzy model was optimized by its two stages of learning: (1) by the traditional learning rule of ANFIS during 100 epochs, and (2) by the additional optimization of conjunction operations and consequent parameters according to some version of the learning rule of ANFIS-C1. IP_G indicates the improvement performed by the ANFIS-C2 when compared to the initial Sugeno model, and it exhibits the cooperation of both learning rules for tuning the different kinds of parameters of the fuzzy system. The second Improvement Index, IP_C, is different from IP_G due to the $Initial_{RMSE2}$ obtained from the trained model after the traditional hybrid learning rule of ANFIS (first stage of proposed learning rule). IP_C evaluates the improvement of ANFIS-C2 over the traditional ANFIS. IP_C helps to view ANFIS as an independent initial model.

We should note that, when an ANFIS-CX architecture contains operations that do not include product t-norms, it is possible that the $Final_{RMSE}$ can be greater than its $Initial_{RMSE}$. In that case, the IP generates a negative value, and therefore there is not an improvement. Thus, a negative IP indicates that a fuzzy model with product conjunction operation has better approximation than the proposed and tested ANFIS-CX.

To depict the performance of the proposed models, we present two examples of the function approximation problem. For example 1, we consider the well-known *sinc* function and in example 2 we consider four different target systems composed of a different number of *sinc* functions.

In this comparative analysis, we only take as reference the error coming from the traditional ANFIS, due to its superiority (shown in different works) in function approximation problems versus other approaches, such as MLP, linear regression, among others [6–11]. Also in [7], it was shown that ANFIS with one training epoch (which is equivalent to a Sugeno trained by LSE) gives better approximations than some traditional neural network models trained by the best learning algorithms found in the literature.

For all tests performed in this section, the data training set was created by sampling 121 data pairs getting from the target function, the points obtained from evenly spaced of the input space $[-10,\ 10] \times [-10, 10]$, that is, with $x_1, x_2 = \{-10, -8\ -6\ -4\ -2\ 0\ 2\ 4\ 6\ 8\ 10\}$ values.

The values presented in the next tables are the results obtained from the average of 30 executions of each model. In all tests the configuration parameters of the DEA, *CR* and *F* are equals 0.7, these values are taken from the suggested results in [39]. *NP* and G_{max} parameter values depended on the version of the learning rule.

6.1. Example 1. approximation of the sinc function

As the baseline of the study, the two-dimensional sinc function considered in [7] is used as the target system. The sinc function for a single variable is denoted as:

$$y = sinc(x) = \begin{cases} \dfrac{\sin(x)}{x} & x \neq 0 \\ 1 & otherwise \end{cases} \tag{24}$$

Then, the two-dimensional *sinc* function can be expressed as follows:

$$y = sinc\ (x_1, x_2) = sinc(x_1)sinc(x_2) \tag{25}$$

The training data is obtained taking some samples of Eq. (25) according to the indicated above. The surface plot obtained from

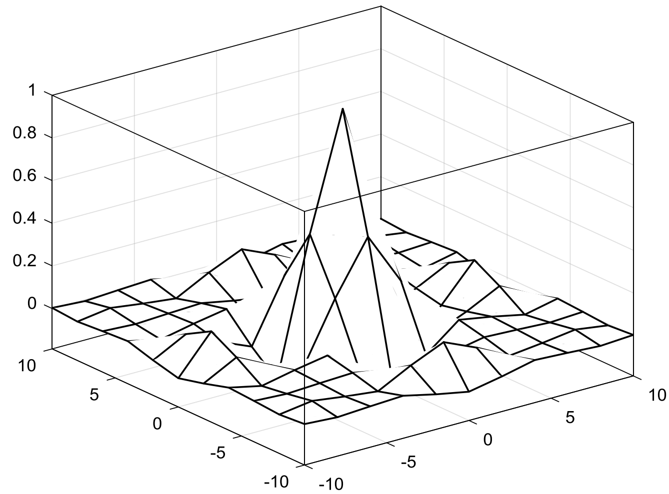

Fig. 8. Surface of the training data set got from the sinc(x,y) target function.

Table 3
$Final_{RMSE1}$ from ANFIS-C1-V1 models for each conjunction operation of the method 4T.

Operation type	Number and type of fuzzy sets					
	Bell(3)	Bell(4)	Bell(5)	Tri(3)	Tri(4)	Tri(5)
DDDL	1.15e−1	1.01e−1	5.50e−2	**8.34e−2**	1.06e−1	4.94e−2
DDDP	1.15e−1	1.03e−1	2.57e−2	**8.34e−2**	9.50e−2	2.58e−2
DDDM	1.15e−1	1.02e−1	2.75e−2	**8.34e−2**	1.04e−1	**2.44e−2**
DLLL	1.15e−1	1.02e−1	5.54e−2	**8.34e−2**	1.07e−1	4.98e−2
DLLP	1.15e−1	1.03e−1	2.57e−2	**8.34e−2**	9.50e−2	2.58e−2
DLLM	1.15e−1	1.02e−1	2.75e−2	**8.34e−2**	1.04e−1	**2.44e−2**
DPPP	1.15e−1	9.49e−2	2.33e−2	**8.34e−2**	8.39e−2	2.58e−2
DPPM	1.15e−1	9.91e−2	2.59e−2	**8.34e−2**	7.77e−2	**2.44e−2**
DMMM	**1.12e−1**	9.46e−2	**2.01e−2**	**8.34e−2**	8.52e−2	**2.44e−2**
LLLP	1.20e−1	1.02e−1	2.57e−2	9.03e−2	9.50e−2	2.58e−2
LLLM	1.21e−1	1.02e−1	2.75e−2	9.93e−2	1.04e−1	**2.44e−2**
LPPP	1.17e−1	8.53e−2	2.33e−2	8.86e−2	7.95e−2	2.58e−2
LPPM	1.19e−1	8.90e−2	2.59e−2	9.24e−2	**7.48e−2**	**2.44e−2**
LMMM	**1.12e−1**	**8.37e−2**	**2.01e−2**	9.19e−2	8.10e−2	**2.44e−2**
PPPM	1.20e−1	1.04e−1	2.61e−2	9.03e−2	9.11e−2	**2.44e−2**
PMMM	1.14e−1	1.02e−1	2.07e−2	8.86e−2	8.33e−2	**2.44e−2**

the training data set is shown in Fig. 8, and it will be approximated by the ANFIS-CX models using different learning rules.

Table 3 shows the $Final_{RMSE1}$ of different configurations of ANFIS-C1-V1. The granularity number and the type of fuzzy sets are specified in the columns, whereas the type of parametric conjunction operations used are specified in the rows. The DEA was set up for 2000 generations (Gmax) and a population size NP = 10.

From the results presented in Table 3, we can note that ANFIS-C1 with *DMMM* and *LMMM* operations have a better approximation in four of the 6-different models. On the other hand, generally, in consistence with the sensitivity analysis given in Section 5, when an ANFIS-C1 uses triangular membership functions, then the approximation capability of the model is better.

The improvement percentage indexes for the best-approximated models are shown in Table 4. Columns 2 and 3 show the $Initial_{RMSE1}$ and $Final_{RMSE1}$, column 4 indicates the IP_1, that is, the improvement performed by ANFIS-C1-V1 to the initial Sugeno model. Column 5 depicts which parametric conjunction operations in model produce the lowest $Final_{RMSE1}$ for the different cases, according to the type and number of input membership functions.

We can highlight that in three of the six models, the IP_1 was greater than 20% for the best conjunction operations. These

Table 4
The Improvement Percentage Indexes for the best ANFIS-C1-V1 models.

Number and type of sets	Initial $RMSE_1$	Final $RMSE_1$	IP_1 (%)	Operations
Bell(3)	1.20e−1	1.12e−1	6.67	DMMM, LMMM
Bell(4)	1.05e−1	8.37e−2	20.29	LMMM
Bell(5)	2.61e−2	2.01e−2	22.99	DMMM, LMMM
Tri(3)	9.03e−2	8.34e−2	7.64	DDDL, DMMM, etc[a]
Tri(4)	9.50e−2	7.48e−2	21.26	LPPM
Tri(5)	2.58e−2	2.44e−2	5.43	DMMM, LMMM, etc[a]

[a] See Table 3 for finding other best conjunctions.

Table 5
$Final_{RMSE2}$ from ANFIS-C2-V1 models for each operation from method 4T.

Operation type	Number and type of fuzzy sets					
	Bell(3)	Bell(4)	Bell(5)	Tri(3)	Tri(4)	Tri(5)
DDDL	1.33e−1	3.34e−2	3.91e−2	7.94e−2	4.33e−2	4.07e−2
DDDP	**3.66e−2**	**1.56e−2**	5.86e−3	7.94e−2	**3.71e−2**	5.87e−3
DDDM	9.48e−2	3.17e−2	9.88e−3	7.94e−2	4.48e−2	1.43e−2
DLLL	1.33e−1	5.26e−2	3.91e−2	7.90e−2	4.76e−2	4.06e−2
DLLP	3.70e−2	1.61e−2	5.86e−3	7.90e−2	**3.71e−2**	5.87e−3
DLLM	7.78e−2	4.96e−2	9.88e−3	7.90e−2	4.76e−2	1.43e−2
DPPP	3.71e−2	1.61e−2	**5.82e−3**	**7.67e−2**	3.74e−2	5.88e−3
DPPM	8.09e−2	4.75e−2	5.90e−3	**7.67e−2**	4.83e−2	8.65e−3
DMMM	9.19e−2	3.97e−2	1.12e−2	**7.67e−2**	4.89e−2	1.43e−2
LLLP	3.70e−2	1.61e−2	5.86e−3	8.31e−2	**3.71e−2**	5.87e−3
LLLM	7.78e−2	4.96e−2	9.88e−3	9.78e−2	5.76e−2	1.43e−2
LPPP	3.71e−2	1.61e−2	**5.82e−3**	8.42e−2	3.74e−2	5.88e−3
LPPM	8.09e−2	4.92e−2	5.90e−3	9.00e−2	5.68e−2	8.65e−3
LMMM	9.19e−2	5.17e−2	1.12e−2	9.02e−2	6.02e−2	1.43e−2
PPPM	3.71e−2	1.59e−2	**5.82e−3**	8.45e−2	3.73e−2	**5.85e−3**
PMMM	3.71e−2	**1.56e−2**	**5.82e−3**	8.45e−2	3.72e−2	5.88e−3

results are very good considering that ANFIS-C1-V1 adds only one conjunction parameter in each model.

Similarly, to the previous test, we approximate the target function using ANFIS-C2 models using the learning rule version 1. Table 5 shows the $Final_{RMSE2}$ for the different configurations studied, and the IP_C and IP_G of the best parametric conjunction operations are shown in Table 6. In the first stage of the composed hybrid learning rule, the ANFIS-C2-V1 was trained with 100 epochs. Moreover, in the second one, the DEA parameters, G_{max} and NP, were set up with the same values as in the previous test. From Tables 3 and 5, it can be noted, that in general, a $Final_{RMSE2}$ is less than $Final_{RMSE1}$ in the corresponding models.

Table 6 shows that the IP_G, which indicates the cooperative learning reaches up to 85.14% for the best case. However, in general, from the information presented in columns IP_C and IP_G, it is clear that most of the improvement is done in the first stage, that is, the improvement done by the hybrid learning rule of ANFIS.

We assumed that the poor improvement carried out by the second stage of the composed hybrid learning rule, seen in the IP_C column, is because there is a unique parameter to train. Additionally, it should understand, IP_C measures the improvement achieved by using conjunction operators in an ANFIS model.

The next test is performed on the ANFIS-C1-V2 model. ANFIS-C1-V2 uses a unique parametric conjunction operation in all rules, but it has the possibility of using different parameter values by rule. DEA was set up for G_{max} = 10,000 generations and a population size NP = 16. Table 7 shows the $Final_{RMSE1}$ of different configurations of ANFIS-C1-V2 and Table 8 shows the IP_1 indexes of the best-approximated models.

Comparing the results from Tables 3 and 7, that is, the $Final_{RMSE1}$ obtained from the optimized ANFIS-C1 models with version 1 and 2 respectively, we can observe that in general, IP_1 has grown significantly. For example, it grew from almost 23% to 67% in

Table 6
The Improvement Percentage Indexes for the best ANFIS-C2-V1 models.

Number and type of sets	Initial $RMSE_1$	Initial $RMSE_2$	Final $RMSE_2$	IP_C (%)	IP_G (%)	Operations
Bell(3)	1.20e−1	3.72e−2	**3.66e−2**	1.61	69.50	DDDP
Bell(4)	1.05e−1	1.63e−2	**1.56e−2**	4.29	85.14	DDDP, PMMM
Bell(5)	2.61e−2	5.87e−3	**5.82e−3**	0.85	77.70	DPPP, LPPP, PPM; PMMM
Tri(3)	9.03e−2	8.45e−2	**7.67e−2**	9.23	15.06	DPPP, DPPM, DMMM
Tri(4)	9.50e−2	3.74e−2	**3.71e−2**	0.80	60.95	DDDP, DLLP, LLLP
Tri(5)	2.58e−2	5.88e−3	**5.85e−3**	0.51	77.33	PPPM

Table 7
$Final_{RMSE1}$ from ANFIS-C1-V2 models for different conjunction operation from method 4T.

Operation type	Number and type of fuzzy sets					
	Bell(3)	Bell(4)	Bell(5)	Tri(3)	Tri(4)	Tri(5)
DDDL	1.12e−1	1.00e−1	4.92e−2	8.21e−2	1.04e−1	4.47e−2
DDDP	1.11e−1	6.42e−2	2.41e−2	8.02e−2	5.72e−2	2.08e−2
DDDM	1.10e−1	5.99e−2	2.08e−2	8.07e−2	7.44e−2	2.23e−2
DLLL	1.12e−1	7.76e−2	4.93e−2	8.17e−2	6.79e−2	4.47e−2
DLLP	1.11e−1	5.23e−2	2.43e−2	7.99e−2	5.46e−2	2.06e−2
DLLM	1.10e−1	4.93e−2	2.05e−2	8.05e−2	5.96e−2	2.24e−2
DPPP	1.10e−1	5.67e−2	2.19e−2	7.82e−2	5.61e−2	1.96e−2
DPPM	1.10e−1	4.19e−2	1.94e−2	8.12e−2	4.83e−2	1.91e−2
DMMM	**1.06e−1**	**3.48e−2**	**1.28e−2**	**7.74e−2**	4.83e−2	2.21e−2
LLLP	1.19e−1	8.85e−2	2.47e−2	8.73e−2	5.40e−2	2.17e−2
LLLM	1.18e−1	8.24e−2	2.10e−2	9.84e−2	6.06e−2	2.14e−2
LPPP	1.17e−1	8.20e−2	2.24e−2	8.45e−2	5.34e−2	1.89e−2
LPPM	1.18e−1	8.16e−2	1.96e−2	8.06e−2	5.45e−2	1.87e−2
LMMM	1.10e−1	8.06e−2	1.34e−2	8.16e−2	**4.63e−2**	2.09e−2
PPPM	1.18e−1	9.71e−2	1.97e−2	8.21e−2	6.52e−2	**1.72e−2**
PMMM	1.13e−1	9.68e−2	1.79e−2	8.17e−2	4.79e−2	2.01e−2

Table 8
IP_1 for the best ANFIS-C1-V2 models.

Number and type of sets	Initial $error_1$	Final $error_1$	IP_1 (%)	Operations
Bell(3)	1.20e−1	**1.06e−1**	11.67	DMMM
Bell(4)	1.05e−1	**3.48e−2**	66.86	DMMM
Bell(5)	2.61e−2	**1.28e−2**	50.96	DMMM
Tri(3)	9.03e−2	**7.74e−2**	14.29	DMMM
Tri(4)	9.50e−2	**4.63e−2**	51.26	LMMM
Tri(5)	2.58e−2	**1.72e−2**	33.33	PPPM

Table 9
$Final_{RMSE1}$ (RMSE) from ANFIS-C2-V2 models for different operations from method 4T.

Operation type	Number and type of fuzzy sets					
	Bell(3)	Bell(4)	Bell(5)	Tri(3)	Tri(4)	Tri(5)
DDDL	1.33e−1	1.96e−2	3.90e−2	7.21e−2	3.02e−2	4.02e−2
DDDP	**3.59e−2**	1.47e−2	4.79e−3	**5.55e−2**	3.20e−2	5.31e−3
DDDM	6.46e−2	1.99e−2	6.81e−3	6.11e−2	**2.16e−2**	1.07e−2
DLLL	1.32e−1	4.05e−2	3.89e−2	7.88e−2	4.30e−2	3.96e−2
DLLP	3.64e−2	1.48e−2	4.74e−3	7.83e−2	3.47e−2	4.79e−3
DLLM	5.28e−2	3.56e−2	7.83e−3	7.83e−2	3.24e−2	1.10e−2
DPPP	3.67e−2	1.29e−2	4.64e−3	7.49e−2	3.63e−2	**3.79e−3**
DPPM	4.29e−2	2.98e−2	**3.90e−3**	7.56e−2	4.45e−2	5.20e−3
DMMM	4.28e−2	3.12e−2	9.51e−3	7.26e−2	4.05e−2	9.40e−3
LLLP	3.64e−2	1.50e−2	5.30e−3	7.91e−2	3.71e−2	5.22e−3
LLLM	5.22e−2	3.71e−2	5.77e−3	9.60e−2	5.36e−2	1.04e−2
LPPP	3.67e−2	**1.24e−2**	5.76e−3	8.05e−2	3.67e−2	4.89e−3
LPPM	4.20e−2	3.07e−2	4.39e−3	7.97e−2	4.79e−2	6.11e−3
LMMM	3.85e−2	3.60e−2	8.26e−3	8.19e−2	5.13e−2	1.02e−2
PPPM	3.71e−2	1.53e−2	5.42e−3	7.84e−2	3.09e−2	5.30e−3
PMMM	3.70e−2	1.43e−2	5.52e−3	7.74e−2	3.14e−2	4.97e−3

the best case. The improvement accomplished by the ANFIS-C1-V2 models was it was expected, because they have 9, 16 or 25 conjunction parameters according to the granularity of the model.

We repeat the previous test, but now, we use the learning rule version 2 in ANFIS-C2 models. The DEA parameters G_{max} and NP have the same values as in the previous test. Table 9 shows the $Final_{RMSE2}$ and Table 10 shows IPs values achieved respectively in this test. The IP_G obtained from the best approximation is equal to 88.19%. According to the IP_C column, it is evidenced that the improvement due to stage 2 of the learning rule was more significant than in the case of using learning rule version 1.

From the results of Tables 4, 6, 8 and 10, we can observe that if the model has more conjunction parameters, that is, changing the learning rule from version 1 to version 2, we can achieve a better improvement on results. However, we can note from IP_1 and IP_C columns that the improvement accomplished by the tuning of conjunction and consequent parameters is less than that made by traditional ANFIS.

Table 11 shows the test of ANFIS-C1 with learning rule version 3. Here the ANFIS-C1 can use any parametric conjunction operation from the method 4T with independent p value. To begin the test, we use the same initial Sugeno models used in the test showed in Table 3. DEA was set up for $G_{max} = 15000$ generations and a population size NP = 16. We can observe that the IP_1s obtained in this test are better than the ones in versions 1 and 2. However, the improvements accomplished by ANFIS-C1-V3 are still a little less than the ones accomplished by traditional ANFIS, if we compare the improvement performed by the tuning of antecedent parameters versus conjunction parameters.

Finally, for this example, we apply the learning rule version 3 to the ANFIS-C2 model. Table 12 shows the results for training ANFIS with different conjunction operations generated by the 4T method. DEA parameters G_{max} and NP have the same values as in the previous test. From the results, it can be noted that the cooperative improvement in the approximation capability up to 80% in the best case. From the IP_C and IP_G columns, we can conclude that the improvement performed in each stage of composed hybrid learning rule have similar magnitudes.

Table 12 also shows two additional columns, the IP_A column indicates the improvement performed by ANFIS to initial Sugeno model and, the column IP_{gr} indicates the improvement performed to increase in one unit the granularity of the model, i.e., incrementing by 1 the number of each input fuzzy sets. These columns are shown for having as references the improvement of a model to change other elements that belong to it. With the presented results, we can conclude that the improvement achieved by training conjunction parameters, using the learning rule version 3, has a similar improvement percentage index than the improvement achieved by the tuning of other elements of the model.

6.2. Example 2. Approximation of multi-sinc functions using ANFIS-C2 models

The *multi-sinc* target functions, used to test the improvement of ANFIS-CX, are generated by the sum of shifted-sinc functions. The mathematical functions analyzed in this example were selected arbitrarily. The objective of the study of this function is to test the proposed models in complex systems. Functions are defined as follows:

$$y = sinc_2\left(x_1, x_2\right) = sinc\left(x_1, x_2\right) + sinc\left(x_1 + 4, x_2 + 2\right) \tag{26a}$$

Table 10
The improvement percentage indexes for the best ANFIS-C2-V2 models.

Number and type of sets	Initial $Error_1$	Initial $Error_2$	Final $Error_G$	IP_C (%)	IP_G (%)	Operations
Bell(3)	1.20e−1	3.72e−2	**3.59e−2**	3.49	70.08	DDDP
Bell(4)	1.05e−1	1.78e−2	**1.24e−2**	30.34	88.19	LPPP
Bell(5)	2.61e−2	5.87e−3	**3.90e−3**	33.56	85.06	DDDL
Tri(3)	9.03e−2	8.45e−2	**5.55e−2**	30.53	34.99	DDDP
Tri(4)	9.50e−2	3.74e−2	**2.16e−2**	42.25	77.26	DDDM
Tri(5)	2.58e−2	5.88e−3	**3.79e−3**	32.48	84.61	DPPP

Table 11
$Final_{RMSE1}$ and IP_1 indexes of ANFIS-C1-V3 models.

Number and type of sets	Initial $RMSE_1$	Final $RMSE_1$	IP_1 (%)
Bell(3)	1.20e−1	1.06e−1	11.67
Bell(4)	1.05e−1	3.10e−2	70.48
Bell(5)	2.61e−2	1.26e−2	51.72
Tri(3)	9.03e−2	7.71e−2	14.62
Tri(4)	9.50e−2	3.62e−2	61.89
Tri(5)	2.58e−2	1.64e−2	36.43

$$y = sinc_3(x_1, x_2) = sinc_2(x_1, x_2) + sinc(x_1 - 5, x_2 - 4) \tag{26b}$$

$$y = sinc_4(x_1, x_2) = sinc_3(x_1, x_2) + sinc(x_1 + 2, x_2 - 6) \tag{26c}$$

$$y = sinc_5(x_1, x_2) = sinc_4(x_1, x_2) + sinc(x_1, x_2 + 8) \tag{26d}$$

The explicit definition of $sinc_2(x, y)$ function is given as follows:

$$y = sinc(x_1)\, sinc(x_2) + sinc(x_1 + 4)\, sinc(x_1 + 2) \tag{27}$$

The rest of multi-sinc functions can be expressed explicitly in a direct way. Fig. 9 shows the complex surfaces of multi-sinc functions for $n_s = 2, 3, 4$ and 5 multi-sinc functions. When $n_s = 1$, then the target function is the same that in example 1.

In this example, we only study the case where the initial fuzzy model has five triangular membership functions for each input. For the learning rules version 2 and 3, the ANFIS-C2 models use PPPM conjunction operations in all of their rules. The values of the parameters configuration for the DEA were the same as the used in example 1 in the respective version of the learning rules.

Table 13 shows the results of the approximations of multisinc functions when ANFIS-C2 is implemented and trained by the different versions of composed hybrid learning rules. In this example, the initial RMSE, $Initial_{RMSE2}$, was obtained from the RMSE of traditional ANFIS. The IP_c columns show only the improvement performed by the second step of the composed hybrid learning rules. The IP_G and the IP_A have similar magnitude than example 1, and the reader can verify it.

We can conclude with this example, that the incorporation of parametric conjunction operations into a fuzzy model (specifically an ANFIS model) that has been trained previously by some intelligent technique, can improve its approximation capability.

Also, we want to highlight that in ANFIS-C2-V3, the additional improvement in the approximation capability to apply second stage of composed hybrid learning rule has similar magnitude than when a fuzzy model adds granularity in its inputs, and it was trained by traditional ANFIS, independently of the complexity of the target function.

Additionally, we want to state that in this work, we analyzed the improvement obtained in the approximation capability, when parametric conjunction operations are added to a fuzzy system. However, our study is limited to one class of parametric conjunction operations that are adequate for embedded systems or hardware implementations [22,37,43,44].

Finally, we want to highlight the next obvious relation was satisfied, a fuzzy system with more parameters, the approximation capability can be improved but the generations executed (and time execution) to find optimal values increase.

For this reason, this study can be extended to the analysis of fuzzy systems using both: (a) other simple parametric conjunction operations adequate to hardware implementation; and (b) other complex parametric conjunction operations [19]. On the other hand, we pave the way to the analysis of the use of other metaheuristics and learning rules for finding the optimal values of the different classes of parameters in the ANFIS-CX architectures or another kind of fuzzy systems.

7. Conclusions

In this paper, we have shown that the incorporation and optimization of parametric conjunction operations in fuzzy models can significantly improve the performance of fuzzy systems. We designed and tested two architectures of fuzzy models: (a) the ANFIS-C1 architecture with parametric operations in conjunction and output linear functions and, (b) the ANFIS-C2 architecture with parametric fuzzy sets additional to parametric operation of the other architecture. Both architectures have the possibility of tuning the conjunction operation parameters using some constraining levels: (i) using the same parametric conjunction operation for all the rules; (ii) using the same parametric conjunction operation with the possibility of having independent parameter value in each rule and; (iii) using both independent parametric conjunction operations and parametric value in each rule.

ANFIS-C1 is useful in the cases where it is desirable to fix input fuzzy sets due to the need to retain the knowledge embedded in them or because they have been optimized by other computational intelligent techniques. On the other hand, the ANFIS-C2 architecture can be used when we wish to tune all parametric fuzzy operations of the fuzzy system. The architectures and optimization methods proposed are more adequate for collaborative instead of competitive with other optimization methods

The learning rule proposed in this paper is motivated in the hybrid learning rule used in traditional ANFIS. By the separation of the antecedent, conjunction and consequent parameters, we can reduce the search space dimensions and optimizing a system in different stages. Additionally, it takes advantage of training linear parameters by LSE and finding optimal consequent parameters.

In addition, in this paper, it was also introduced to the sensitivity regions concept, which does a reference to the local input regions where the parametric conjunction operations influence on the overall output. It was shown, that these regions have a relationship with the shape of the input membership function.

With the results presented in this paper, we showed and highlighted that the approximation capability, achieved by adding parametric conjunction operations to a fuzzy system, can be of similar magnitude to the cases when the parameters of fuzzy sets are tuned or when the granularity of the system is increased.

In future works, we will perform studies of the approximation capability of the models presented in this work using other types

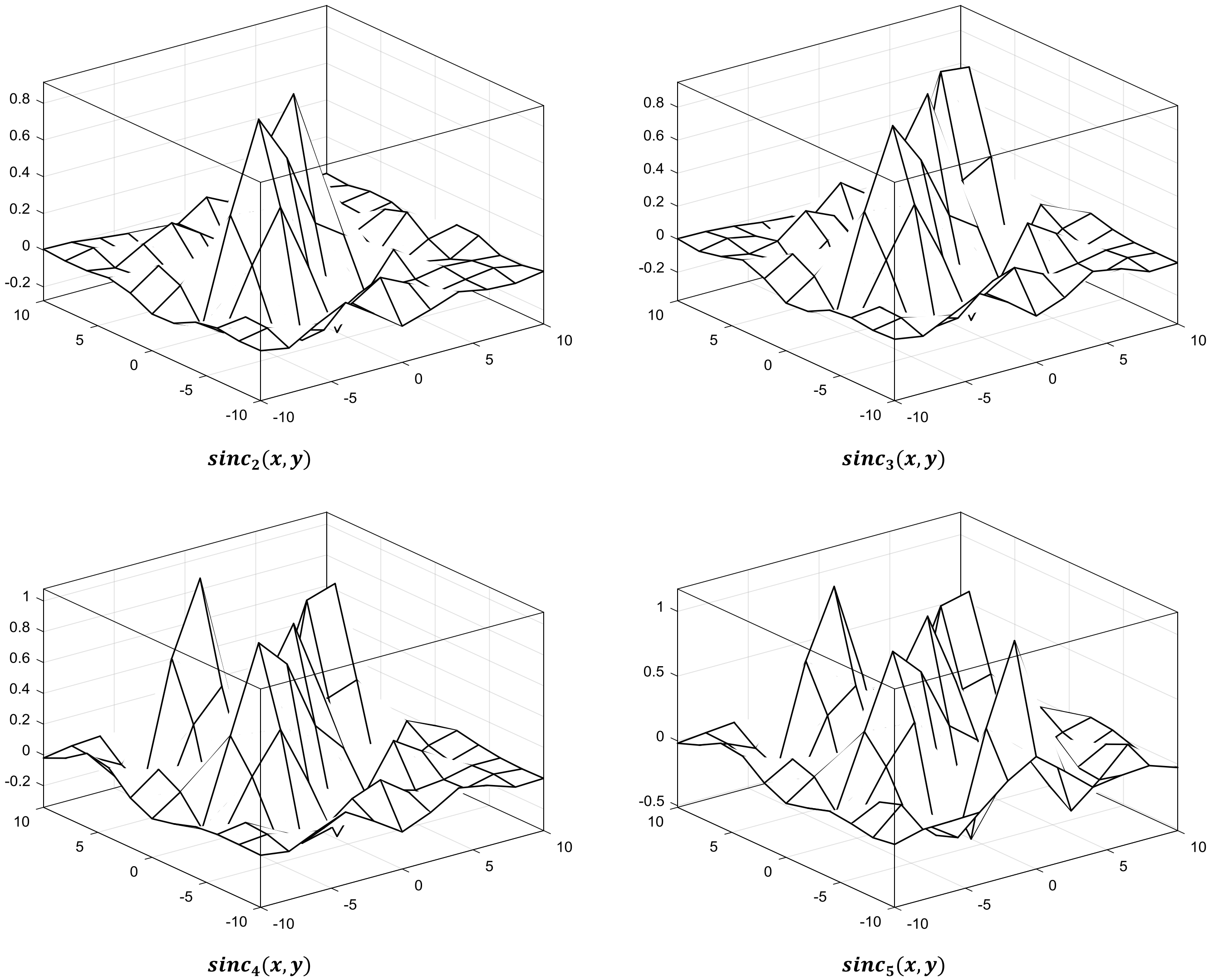

Fig. 9. Surfaces of shifted-sinc functions.

Table 12
$Final_{RMSE2}$ and IP indexes of ANFIS-C2-V3 models.

Number and type of sets	Initial $RMSE_1$	Initial $RMSE_2$	Final $Error_2$	IP_C (%)	IP_G (%)	IP_A (%)	IP_{gr} (%)
Bell(3)	1.20e−1	3.72e−2	3.55e−2	4.57	70.42	69.00	–
Bell(4)	1.05e−1	1.78e−2	1.14e−2	35.96	89.14	84.48	12.5
Bell(5)	2.61e−2	5.87e−3	3.22e−3	45.14	87.66	77.51	78.25
Tri(3)	9.03e−2	8.45e−2	4.98e−2	41.07	44.85	6.42	–
Tri(4)	9.50e−2	3.74e−2	1.90e−2	49.20	80.00	60.63	−5.20
Tri(5)	2.58e−2	5.88e−3	2.68e−3	54.42	89.61	77.21	71.43

Table 13
Optimization results of ANFIS-C2 by different learning rules from the 4T method to approximate muli-sinc functions.

Number of sinc functions	ANFIS-C2-V1			ANFIS-C2-V2		ANFIS-C2-V3	
	Initial $RMSE_2$	Final $Error_2$	IP_c	Final $Error_2$	IP_c	Final $Error_2$	IP_c
1	5.88e−3	5.85e−3	0.51	3.79e−3	35.54	2.52e−3	57.14
2	9.73e−3	9.70e−3	0.31	6.26e−3	35.66	5.06e−3	48.00
3	9.80e−3	9.74e−3	0.61	6.95e−3	29.08	4.77e−3	51.33
4	1.41e−2	1.37e−2	2.84	7.69e−3	45.46	4.93e−3	65.04
5	2.90 e-2	2.60 e-2	10.34	1.73e−2	40.34	1.38e−2	52.41

of parametric conjunction operations more suitable for hardware realization. In this work, we also pave the way to the application of other metaheuristics for optimizing parametric conjunction operations and for using parametric conjunction operations in fuzzy models with applications in the control and robotics areas.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106095.

CRediT authorship contribution statement

Prometeo Cortés-Antonio: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Project administration. **Ildar Batyrshin:** Conceptualization, Methodology, Software, Formal analysis, Resources, Writing - original draft, Visualization, Supervision, Project administration. **Alfonso Martínez-Cruz:** Software, Investigation, Writing - original draft, Writing - review & editing. **Luis A. Villa-Vargas:** Validation, Data curation, Writing - review & editing, Supervision. **Marco A. Ramírez-Salinas:** Validation, Investigation, Data curation, Writing - review & editing, Supervision. **Imre Rudas:** Conceptualization, Resources, Writing - review & editing, Supervision. **Oscar Castillo:** Methodology, Validation, Formal analysis, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Herón Molina-Lozano:** Conceptualization, Software, Writing - review & editing, Supervision, Project administration.

Acknowledgments

The authors would like to express our thankful to CONACYT, Mexico, CIC-IPN, Mexico and Tijuana Institute of Technology, Mexico by their support to this research work.

References

[1] A. Ali-Eldin, J. Van den Berg, H.A. Ali, A risk evaluation approach for authorization decisions in social pervasive applications, Comput. Electr. Eng. 55 (2016) 59–72.
[2] K. Ashish, A. Dasari, S. Chattopadhyay, N.B. Hui, Genetic-neuro-fuzzy system for grading depression, Appl. Comput. Inform. 14 (1) (2017) 98–105.
[3] S. Kar, S. Das, P.K. Ghosh, Applications of neuro fuzzy systems: A brief review and future outline, Appl. Soft Comput. 15 (2014) 243–259.
[4] M. Morshedizadeh, M. Kordestani, R. Carriveau, D.S.-K. Ting, M. Saif, Application of imputation techniques and adaptive neuro-fuzzy inference system to predict wind turbine power production, Energy 138 (2017) 394–404.
[5] J. Yen, R. Langari, L.A. Zadeh, Industrial Applications of Fuzzy Logic and Intelligent Systems, IEEE Press, 1995.
[6] J.-S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (3) (1993) 665–685.
[7] J.-S.R. Jang, C.T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing; A Computational Approach To Learning and Machine Intelligence, Prentice Hall, 1997.
[8] A.E. Bakyani, H. Sahebi, M.M. Ghiasi, N. Mirjordavi, F. Esmaeilzadeh, M. Lee, A. Bahadori, Prediction of CO2–oil molecular diffusion using adaptive neuro-fuzzy inference system and particle swarm optimization technique, Fuel 181 (2016) 178–187.
[9] L. Naderloo, H. Javadikia, M. Mostafa, Modeling the energy ratio and productivity of biodiesel with different reactor dimensions and ultrasonic power using ANFIS, Renew. Sustain. Energy Rev. 70 (2017) 56–64.
[10] Y. Tan, C. Shuai, L. Jiao, L. Shen, An adaptive neuro-fuzzy inference system (ANFIS) approach for measuring country sustainability performance, Environ. Impact Assess. Rev. 65 (2017) 29–40.
[11] Y. Tao, Y. Wang, M. Pan, S. Zhong, Y. Wu, R. Yang, Y. Han, J. Zhou, Combined ANFIS and numerical methods to simulate ultrasound-assisted extraction of phenolics from chokeberry cultivated in China and analysis of phenolic composition, Sep. Purif. Technol. 178 (2017) 178–188.
[12] A. Zendehboudi, X. Li, B. Wang, Utilization of ANN and ANFIS models to predict variable speed scroll compressor with vapor injection, Int. J. Refrig. 74 (2017) 475–487.
[13] A. Đokic, S. Jovic, Evaluation of agriculture and industry effect on economic health by ANFIS approach, Physica A 479 (2017) 396–399.
[14] W. Suparta, K.M. Alhasa, Modeling of Tropospheric Delays using ANFIS, Springer, 2016.
[15] I. Batyrshin, O. Kaynak, Parametric classes of generalized conjunction and disjunction operations for fuzzy modeling, IEEE Trans. Fuzzy Syst. 7 (5) (1999) 586–596.
[16] I. Batyrshin, O. Kaynak, I.J. Rudas, Fuzzy modeling based on generalized conjunction operations, IEEE Trans. Fuzzy Syst. 10 (5) (2002) 678–683.
[17] J. Qiao, W. Li, X.-J. Zeng, H. Han, Identification of fuzzy neural networks by forward recursive input–output clustering and accurate similarity analysis, Appl. Soft Comput. 49 (2016).
[18] M.D. Ruiz, J. Gómez-Romero, M. Molina-Solana, J.R. Campaña, M.J. Martin-Bautista, Meta-association rules for mining interesting associations in multiple datasets, Appl. Soft Comput. 49 (2016) 212–223.
[19] E.P. Klement, R. Mesiar, E. Pap, Triangular Norms, Springer, 2000.
[20] G. Mayor, J. Torrens, Triangular norms on discrete settings, in: Logical, Algebraic, Analytic and Probabilistic Aspects of Triangular Norms, 2005, pp. 189–230.
[21] I. Batyrshin, I.J. Rudas, L.A. Villa-Vargas, P. Cortés-Antonio, On the monotone sum of basic t-norms in the construction of parametric families of digital conjunctors for fuzzy systems with reconfigurable logic, Knowl.-Based Syst. 38 (2013) 27–36.
[22] A. Henández-Zavala, I. Batyrshin, O. Camacho-Nieto, I.J. Rudas, G. Sidorov, L.A. Villa-Vargas, On generation and FPGA implementation of digital fuzzy parametric conjunctions, Appl. Comput. Mathe. 11 (2) (2012) 150–164.
[23] L.N. Stout, Categorical approaches to non-commutative fuzzy logic, Fuzzy Sets and Systems 161 (18) (2010) 2462–2478.
[24] P. Koprinkova-Hristova, Fuzzy operations' parameters versus membership functions' parameters influence on fuzzy control systems properties, in: 2004 2nd International IEEE Conference on'Intelligent Systems'. Proceedings (IEEE Cat. No. 04EX791), vol. 1, 2004, pp. 219–224.
[25] A. Tellez, H. Molina-Lozano, L.A. Villa-Vargas, E. Rubio, I. Batyrshin, Parametric type-2 fuzzy logic systems, in: E.P. Dadios (Ed.), Fuzzy Logic-Algorithms, Techniques and Implementations, 2012, pp. 97–114.
[26] M.M. Gupta, J. Qi, Design of fuzzy logic controllers based on generalized T-operators, Fuzzy Sets and Systems 40 (1991) 473–489.
[27] J. Alcalá-Fdez, F. Herrera, F.A. Márquez, A. Peregrín, Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems, Int. J. Intell. Syst. 22 (9) (2007) 1035–1064.
[28] M. Korytkowski, R. Scherer, Modular neuro-fuzzy systems based on generalized parametric triangular norms, in: International Conference on Parallel Processing and Applied Mathematics, 2009, pp. 332–339.
[29] A.C. Aras, O. Kaynak, R.H. Abiyev, Slip control of a quarter car model based on type-1 fuzzy neural system with parameterized conjunctions, in: IECon 2012-38th Annual Conference on IEEE Industrial Electronics Society, 2012, pp. 2488–2493.
[30] A.C. Aras, O. Kaynak, Trajectory tracking of a 2-DOF helicopter system using neuro-fuzzy system with parameterized conjunctors, in: 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2014, pp. 322–326.
[31] A.C. Aras, O. Kaynak, Fuzzy interval TSK type-2 modeling with parameterized conjunctors, Asian J. Control 17 (3) (2015) 963–976.
[32] R. Lovassy, L.T. Kóczy, L. Gál, Generalization capability of neural networks based on fuzzy operators, Appl. Comput. Math. 10 (2) (2011) 340–355.
[33] R. Lovassy, L.T. Kóczy, L. Gál, Function approximation performance of fuzzy neural networks, Acta Polytech. Hung. 7 (4) (2010) 25–38.
[34] R.B. Nelsen, An Introduction To Copulas, Springer Science & Business Media, 2007.
[35] M. Craiu, R.V. Craiu, On the choice of parametric families of copulas, Adv. Appl. Stat. 10 (1) (2008) 25–40.
[36] A. Rybalov, E. Kagan, R. Yager, Parameterized uninorm and absorbing norm and their application for logic design, in: 2012 IEEE 27th Convention of Electrical & Electronics Engineers, Israel, 2012.
[37] P. Cortés-Antonio, I. Batyrshin, L.A. Villa-Vargas, I.J. Rudas, H. Molina-Lozano, M.A Ramirez-Salinas, Hardware design of digital parametric conjunctors and t-norms, Int. J. Fuzzy Syst. 17 (4) (2015) 559–576.
[38] K. Price, R. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach To Global Optimization, Springer Science & Business Media, 2006.
[39] E. Mezura-Montes, J. Velázquez-Reyes, C.A. Coello-Coello, A comparative study of differential evolution variants for global optimization, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 200, pp. 485–492.
[40] U. Chakraborty, Advances in Differential Evolution, vol. 143, Springer, 2008.
[41] D. Corne, M. Dorigo, F. Glover, New Ideas in Optimization, McGraw-Hill, 1998.
[42] S. Jenei, How to construct left-continuous triangular norms-state of the art, Fuzzy Sets and Systems 143 (1) (2004) 27–45.
[43] P. Cortés-Antonio, I. Batyrshin, H. Molina-Lozano, FPGA implementation of fuzzy mamdani system with parametric conjunctions generated by monotone sum of basic t-norms, Polibits 44 (2011) 53–58.

[44] I. Batyrshin, I.J. Rudas, A. Panova, On generation of digital fuzzy parametric conjunctions, Stud. Comput. Intell. 243 (2009) 79–89.
[45] M. Sugeno, G.T. Kang, Structure identification of fuzzy model, Fuzzy Sets and Systems 28 (1) (1988) 15–33.
[46] K. Mohammadi, S. Shamshirband, D. Petkovic, P.L. Yee, Z. Mansor, Using ANFIS for selection of more relevant parameters to predict dew point temperature, Appl. Therm. Eng. 96 (2016) 311–319.
[47] R. Sharma, M.S. Patterh, A new pose invariant face recognition system using PCA and ANFIS, Optik 126 (23) (2015) 3483–3487.
[48] J. Sobhani, M. Najimi, A.R. Pourkhorshidi, T. Parhizkar, Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models, Constr. Build. Mater. 24 (5) (2010) 709–718.
[49] O. Popoola, J. Munda, A. Mpanda, Comparative analysis and assessment of ANFIS-based domestic lighting profile modelling, Energy Build. 107 (2015) 294–306.
[50] P.M. Vasant, Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance, IGI Global, 2012.
[51] M. Jamshidi, R.A. Krohling, L. d. S. Coelho, P.J. Fleming, Robust Control Systems with Genetic Algorithms, vol. 3, CRC Press, 2002.
[52] A. Martínez-Cruz, R. Barrón-Fernández, H. Molina-Lozano, M.A. Ramírez-Salinas, L.A. Villa-Vargas, P. Cortés-Antonio, K.-T.T. Cheng, An automatic functional coverage for digital systems through a binary particle swarm optimization algorithm with a reinitialization mechanism, J. Electron. Test. 33 (4) (2017) 431–447.
[53] E. Naderi, A. Azizivahed, H. Narimani, M. Fathi, M.R. Narimani, A comprehensive study of practical economic dispatch problems by a new hybrid evolutionary algorithm, Appl. Soft Comput. 61 (2017) 1186–1206.
[54] R. Storn, K. Price, Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[55] J.P. Donate, X. Li, G. Sánchez, A. Sanchis De Miguel, Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm, Neural Comput. Appl. 22 (1) (2013) 11–20.
[56] L. Tang, Y. Zhao, J. Liu, An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production, IEEE Trans. Evol. Comput. 18 (2) (2014) 209–225.
[57] R. Storn, K.V. Price, Minimizing the real functions of the ICEC'96 contest by differential evolution, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1996.
[58] L. Amador-Angulo, O. Castillo, A new fuzzy bee colony optimization with dynamic adaptation of parameters using interval type-2 fuzzy logic for tuning fuzzy controllers, Soft Comput. 22 (2) (2018) 571–594.
[59] F. Olivas, F. Valdez, O. Castillo, C.I. Gonzalez, G. Martinez, P. Melin, Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems, Appl. Soft Comput. 53 (2017) 74–87.
[60] E. Ontiveros-Robles, P. Melin, O. Castillo, New methodology to approximate type-reduction based on a continuous root-finding karnik mendel algorithm, Algorithms 10 (3) (2017) 77.
[61] J. Perez, F. Valdez, O. Castillo, P. Melin, C. Gonzalez, G. Martinez, Interval type-2 fuzzy logic for dynamic parameter adaptation in the bat algorithm, Soft Comput. 21 (3) (2017) 667–685.
[62] R. Malhotra, N. Singh, Y. Singh, Soft computing techniques for process control applications, Int. J. Soft Comput. 2 (3) (2011) 32–44.
[63] K.T. Atanassov, Intuitionistic fuzzy sets, Fuzzy Sets Syst. 20 (1) (1986) 87–96.
[64] J.M. Mendel, Uncertain Rule-Based Fuzzy Systems, Springer, 2017.
[65] O. Castillo, P. Melin, R. Tsvetkov, K.T. Atanassov, Short remark on fuzzy sets, interval type-2 fuzzy sets, general type-2 fuzzy sets and intuitionistic fuzzy sets, in: Intelligent Systems' 2014, Springer, 2015, pp. 183–190.