Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Optimal Design of Adaptive Neuro-Fuzzy Inference System Using PSO And Ant Colony Optimization For Estimation of Uncertain Observed Values

Mahdi Danesh (danesh1980@yahoo.com)
Imam Khomeini International University
Sedighe Danesh
Islamic Azad University East Tehran Branch

Research Article

Keywords: Uncertainty, Fuzzy regression, Linear programming, Machining process, Adaptive neuro-fuzzy inference system, Ant colony, Particle swarm.

Posted Date: October 18th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-666682/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Optimal Design of Adaptive Neuro-Fuzzy Inference System Using PSO and ant Colony Optimization for Estimation of uncertain observed values

Mahdi Danesh[1] * ,Sedigheh Danesh[2]

[1] Imam Khomeini International University – Buin Zahra Higher Education Center of Engineering and Technology, Buein Zahra, Ghazvin, Iran E-mails: danesh@bzte.ac.ir

[2] Islamic Azad University- East Tehran branch, Tehran, Iran

Abstract

This study employs a new method for regression model prediction in an uncertain environment and presents fuzzy parameter estimation of fuzzy regression models using triangular fuzzy numbers. These estimation methods are obtained by new learning algorithms in which linear programming is used. In this study, the new algorithm is a combination of a fuzzy rule-based system, on the basis of particle swarm optimization (PSO) and ant Colony Optimization ACO_R. In addition, a simulation and a practical example in the field of machining process are applied to indicate the performance of the proposed methods in dealing with problems where the observed variables have the nature of uncertainty and randomness. Finally, the results of the proposed algorithms are evaluated.

Keywords: Uncertainty, Fuzzy regression; Linear programming; Machining process; Adaptive neuro-fuzzy inference system; Ant colony; Particle swarm.

1. Introduction

For precise and reliable estimation, it is important to consider uncertainty in the construction of the manufacturing process models which are based on input/output data. As a result of uncertainty, imprecision and randomness associated with the measurement, materials, process parameters, and so on, in most experimental tests, the observed values of the repeated tests with the same inputs on the same samples are not an equal value. Besides, in some cases such as surface roughness resulted from machining, the obtained output values are inherently random.

As a result, the construction of the model is based on the output variables consisting of an average group of the repeated observed values without any exact variation around the mean. Uncertainty was classified by Isukapalli and coauthors (1998) into the following types: Natural Uncertainty, Parameter Uncertainty and Data Uncertainty. Through input-output relations, these uncertainties are spread and combined. When uncertainty quantification techniques differ from each other, two types of general uncertainty can be derived: probabilistic uncertainty and possibilistic (non-probabilistic) uncertainty (Wang 2019). Probabilistic uncertainty usually describes the uncertain parameter when data information is sufficient. However, recent years have witnessed the emergence of the Possibilistic uncertainty approaches, specifically the interval theory and fuzzy set theory to address the issues (Wang 2020, Faes 2020). Fuzzy number approaches which are an extension to interval methods, assign a membership function to all parameter values which are related to a certain interval (Hanss 2005). Recently, the application of fuzzy logic has widely increased in uncertainty quantification for its successful performance. (Wang 2021,Wang 2021, Zhang 2020) . The linear model of fuzzy regression analysis, established by Tanaka *et al.* (1982),

*corresponding author

has enabled the fuzzy system to give the fuzzy output. In the fuzzy linear programming problems, known as the FFLP problems, all the parameters, as well as the variables, are represented by fuzzy numbers (Kumar 2011). In the fuzzy regression model, a fuzzy functional relationship is given between explanatory variables and response variables. Several fuzzy regression techniques have been suggested based on fuzzy least squares (FLS) and mathematical programming methods, such as linear programming (LP) or quadratic programming (QP) that minimize the total spread of the output. The FLS and LP methods were suggested by Diamond (2011) and Tanaka et al. (1982, 1987,1988, and1989), respectively. Several variants of the FLS (Hong 2001) and LP methods (Razzaghnia 2011, Razzaghnia 2015, Fan 2013, Zhang 2018) have been applied for the fuzzy linear regression problem.

Fuzzy inference system can express the uncertain situations in the form of rules. Therefore, it is applied in solving uncertain problems (Hemayatkar 2018 , Hidalgo 2012). Neuro-fuzzy is a hybrid artificial intelligence technique with fuzzy logic and artificial neural networks. One of the most popular of hybrid algorithms is the ANFIS. Jang (1993) proposed an adaptive neural network based on fuzzy inference systems (ANFIS). In the ANFIS, learning ability and relational structure of the artificial neural networks are combined with the decision-making mechanism of the fuzzy logic. ANFIS has a flexible and strong structure. In order to optimize the consequent parameters in the hybrid algorithm of the adaptive neuro-fuzzy inference system, Danesh et al. (2016, 2020) suggested the new hybrid algorithms. Dong *et al.* present d an adaptive optimal fuzzy logic based energy management solution to develop appropriate day-ahead fuzzy rules for real-time energy dispatch adaptively in the presence of operational uncertainties.

In order to reduce the error of the fuzzy regression model, the current study proposed new algorithms that use the fuzzy inference system and meta-heuristic algorithms such as ant colony and particle swarm algorithms to optimize the premise parameters. In addition, linear programming was used for the consequence parameters prediction. Moreover, these algorithms were compared with the method proposed by Danesh (2018) which was based on adaptive fuzzy inference system and linear programming (FWLP). The study showed that the proposed methods had fewer errors than the LP and FWLP methods, and were further verified by the predictions using simulation and practical examples.

This paper includes five sections. Section 1 is the introduction and the Section 2 explains concepts and formulations of the different models. The methodology of the proposed methods is explained in Section 3. To illustrate the procedure more accurately in predicting uncertain outputs of the proposed methods, two examples are given in Section 4. The first example is a simulated example and the second example describes the application of the proposed methods for predicting surface roughness in machining GFRP composites by considering the uncertainty, irregularity and imprecision associated with surface roughness measurement. In section 5, the analysis of the obtained results is discussed.

2. Basic concepts and methods

In this section, we briefly review basic concepts of fuzzy regression, linear programming (LP) in fuzzy regression, adaptive neuro-fuzzy inference system, particle swarm algorithm, and ant colony algorithm respectively.

2.1 Fuzzy regression

The function $f(\mathrm{x})$ is a mapping from x to Y where $\mathrm{x}_j = \left(x_{j0}, x_{j1}, \dots, x_{jp}\right)$ $(j = 1, \dots, n)$ is a p-dimentional vector crisp independent variable and domain is assumed to be $D \subseteq R^p$. Consider the following the fuzzy regression model:

$$Y = f(\mathrm{x})\{+\}\varepsilon = (l(\mathrm{x}), a(\mathrm{x}), r(\mathrm{x}))_{LR}\ \{+\}\varepsilon. \qquad (1)$$

ε represents the regression error with conditional mean zero and variance $\sigma^2(\mathrm{x})$ given x. In this paper, response variable Y has the symmetric triangular fuzzy structure that Y_j can be written as $Y_j = (a_j, \beta_j)$ where a_j and β_j are the center and the spread of a symmetric triangular fuzzy number respectively and $\beta_j = r_j - a_j = a_j - l_j$.

2.2 Linear Programming (LP) in fuzzy regression

In this study, we consider the following fuzzy regression model as:

$$Y_j = p_0 + p_1 x_{j1} + p_2 x_{j2} + \cdots + p_p x_{jp} = P\mathrm{x}_j, \quad j = 1, \dots, n, \tag{2}$$

where n is the number of data points, $\mathrm{x}_j = (1, x_{j1}, x_{j2}, \dots, , x_{jp})$ is a p-dimensional input vector of the independent variables at the j^{th} observation, also, $P = (p_0, p_1, \dots, p_p)$ is a vector of unknown fuzzy parameters and Y_j is the j^{th} observed value of the dependent variables. P can be denoted in vector form as $P = \{b, \alpha\}$ where $b = (b_0, b_1, \dots, b_p)$, $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_p)$, b_i is the center value and α_i is the spread value of p_i, $i = 0, \dots, p$. Also, $Y_j = (a_j, \beta_j)$ is symmetric triangular fuzzy number where a_j and β_j are the center and the spread this number, respectively. Also, the fuzzy regression parameters can be calculated by solving linear programming (LP) model as follows (Hidalgo 2012):

$$\min L = \sum_{\mathrm{j}=1}^{n} \sum_{i=0}^{p} \alpha_i \left|x_{ji}\right| \tag{3}$$

So that, the following two constraints must be established:

$$\sum_{i=0}^{p} b_i \left|x_{ji}\right| - (1-h) \sum_{i=0}^{p} \alpha_i \left|x_{ji}\right| \le a_j - (1-h)\beta_j, \tag{4}$$

$$\sum_{i=0}^{p} b_i \left|x_{ji}\right| + (1-h) \sum_{i=0}^{p} \alpha_i \left|x_{ji}\right| \ge a_j + (1-h)\beta_j, \tag{5}$$

and $\alpha_i \ge 0, \;\; i = 0, \dots, p, \;\; j = 1, \dots, n \, , 0 < h \le 1$.

In this model, the constrains assure that the support of the predicted values from the regression model includes the support of the observed values.

2.3. Particle Swarm Optimization (PSO)

The particle swarm optimization is a meta-heuristic algorithm that was introduced by Eberhart et al. (1995). It is based on the behavior of animals in a flock by mimicking the collective learning of the group, such as birds. Each bird is represented as a particle. To reach the optimal solution, they share the information among themselves. The number of particles and their positions inside the parameter's space is randomly defined to initialize the method. The PSO method has both global and local searching ability with fast convergence and parameter adjustment. It solves the problem of the sensitivity of the iterative method to the initial value and prevents the clustering algorithm from getting into local optimum. For the optimization problem, each particle in the PSO algorithm represents a possible solution. The best solution found by the particle is the best position experienced by the whole group. $\mathrm{x}^{l,tseb}$ and $\mathrm{x}^{g,tseb}$ are individual extremum and global extremum respectively. To create a new population, every particle constantly updates itself through l, $tseb$, g and $tseb$. Also, the whole population comprehensively searches the solution region. The position of the l^{th} particle $(l = 1, 2, \ldots, N)$ can be expressed as x^l and speed as v^l. The positions and velocities of all the particles are iteratively updated. In order to refine the search around the global optima, the "adaptation" concept has been proposed by Eberhart (1995). Furthermore, velocity limit $[v\ min, vmax]$ and positions limit $[xmin, xmax]$ were suggested to assure that the particle does not move beyond the search space boundaries. The new position and velocity at $(t + 1)$ can be calculated by the following equations:

$$\mathrm{v}^l[t+1] = \mathrm{wv}^l[t] + \mathrm{F}^l[t]. \tag{6}$$

$$\mathrm{F}^l[t] = c_1 R_1 \otimes (x^{l,tseb}[t] - x^l[t]) + c_2 R_2 \otimes (x^{g,tseb}[t] - x^l[t])$$

$$\mathrm{x}^l[t+1] = \mathrm{x}^l[t] + \mathrm{v}^l[t+1].$$
(7)

$\mathrm{F}^l[t]$ is objective function. So, w is the inertial weight whose value decreases rapidly in the first iteration. However, its weight decreases slowly after a number of iterations. c_1 and c_2 are constants and called as acceleration factor. Also, R_1 and R_2 are random numbers that change in the interval of (0, 1). Then the individual extremum of each particle and the global extremum of the whole particles can be updated by the equations as follows:

$$x^{l,tseb}[t+1] = \begin{cases} x^l(t+1) & x^l(t+1) \geq x^{l,tseb}[t] \\ x^{l,tseb}[t] & x^l(t+1) \leq x^{l,tseb}[t] \end{cases} \tag{8}$$

$$x^{g,tseb}[t+1] = max x^{l,tseb}[t+1] \qquad l = 1, \ldots, N, g = 1, \ldots, d$$

The general procedure for this scheme is summarized as follows:

Step 1: t = 0 is inputted.

Step 2: The initial values of positions and velocities for all particles are randomly determined.

Step 3: Positions are updated according to Eq. (6).

Step4: The objective function is computed and evaluated at each particle position.

Step 5: $\mathrm{x}^{l,tseb}[t]$ is computed for $l = 1, \ldots, N$.

Step 6: $\mathrm{x}^{g,tseb}[t]$ is computed for $g = 1, \ldots, d$.

Step 7: velocities are updated according to Eq. (7).

Step 8: t = t + 1.

Step 9: If t> maximum, the training of network terminates, otherwise go to 4.

2.4. Ant Colony Optimization (ACO)

Ant colony optimization is a meta-heuristic technique that uses artificial ants to find solutions to combinatorial optimization problems. It uses a discrete structure to determine the solutions. In a discrete structure, each of the decision variables is divided into a specified number of cases in a defined range. With discrete variable space, a limitation is produced for the algorithm which reduces the accuracy of optimality. If you divide the space between the large numbers of decision variables, the accuracy of the solutions will be increased accordingly. Also, such a complex system may reduce accuracy. To fix this problem, generalized ACO to space R was proposed (ACO_R). Thus, the fundamental idea underlying ACO_R is the shift from using a discrete probability distribution to using a continuous one. In R algorithm, the shift to a continuous space is carried out using a probability density function. Soosha and Dorigo [30] proposed Gaussian function in ant colony optimization for a continuous domain. In order to create multi-maximums, the kernel of Gaussian function (the sum of the singular Gaussian functions) is defined as:

$$f_j(\mathrm{x}) = \sum_{l=1}^{k} \omega^l g_j^l(\mathrm{x}) = \sum_{l=1}^{k} \omega^l \frac{1}{\sqrt{2\pi}\sigma_j^l} \exp((-\tfrac{1}{2})(\tfrac{\mathrm{x}-\tau_j^l}{\sigma_j^l})^2. \tag{9}$$

parameters ω^l ،τ_j^l and σ_j^l should be determined during the execution of the algorithm and k is the number of singular Gaussian functions. Decision variables related to l^{th} solution are denoted by $b_1^l, b_2^l, \dots, b_\mathrm{n}^l$. For solution l and j^{th} variable, τ_j^l is considered equal to b_j^l and σ_j^l is assigned as follows:

$$\sigma_j^l = \vartheta \sum_{z=1}^{k} \frac{\left|b_j^z - b_j^l\right|}{k-1}, \quad j = 1, \dots, n. \tag{10}$$

σ_j^l is the average distance between the j^{th} variable of the solution b^l and the j^{th} variable of the other solutions in the archive and k is the size of the archive. ϑ is an adjustment parameter. Large value ϑ reduces convergence speed. The value ϑ affects the long-term memory causing the worse solutions to be deleted, and the new solutions are considered closer to known good solutions (Dastranj 2011). ω^l is a weight associated with solution j, where for every b^l, a weight ω^l is calculated as follows:

$$\omega^l = \frac{1}{qk\sqrt{2\pi}} \exp\left(\frac{-(l-1)^2}{2q^2k^2}\right). \tag{11}$$

that $\omega^1 \geq \omega^2 \geq \cdots \geq \omega^k$.

The poor decisions will be minimized if a very small value is considered for q. Yet with extreme reduction, the assigned value will be trapped in the local optimum. Also, selection probability of solution j can be expressed as follows:

$$P = \frac{\omega^l}{\sum_{z=1}^{k} \omega^z}. \tag{12}$$

The general procedure of ACO_R is summarized as follows:

Step 1: Based on the answer solution b^l for τ_j^l and σ_j^l for the j^{th} variable, a normal random number is generated to create a new solution.

Step 2: It is repeated for variable n and then for each of the N ant.

Step 3: All of the solutions are sorted according to their quality, where saved k better solutions are retained and the rest of the solutions are discarded.

Step 4: Algorithm stop conditions are as follows:

1. The lack of progress. The new solutions to be the same with the previous solutions.

2. The repetitions number. When the number of repetitions reached to the predetermined repetitions number.

3. Reach to a maximum amount. The algorithm continue to converge to a certain amount. Otherwise go to step 1.

2.5. Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS is a famous hybrid technique that combines the adaptive learning capability of ANN along with the intuitive fuzzy logic of human reasoning (Jang 1993). The network structure of ANFIS consists of two parts. Utilizing an optimization algorithm, the premise and consequence parameters are determined. Also during training, the existing input-output data pairs are used (Jang 1993). The structure of ANFIS consists of five layers. To present ANFIS model architecture, we consider four fuzzy if-then rules with two-dimensional input and one output.

Rule 1: If x_1 is A_1 and x_2 is A_3 then $f_1 = p_0^1 + p_1^1 x_1 + p_2^1 x_2$,

Rule 2: If x_1 is A_1 and x_2 is A_4 then $f_2 = p_0^2 + p_1^2 x_1 + p_2^2 x_2$,

Rule 3: If x_1 is A_2 and x_2 is A_3 then $f_3 = p_0^3 + p_1^3 x_1 + p_2^3 x_2$,

Rule 4: If x_1 is A_2 and x_2 is A_4 then $f_4 = p_0^4 + p_1^4 x_1 + p_2^4 x_2$.

Where, x_1, x_2 and $y \in R$ are input and output variables, respectively. A_k's are fuzzy sets and f_k represents system output due to rule R_k $(k = 1,2,3,4)$. In the following, the five layers of the system are explained.

In the first layer, all the nodes are adaptive nodes that generate membership grades of the inputs. The node functions are given by:

$$o_{1,k} = \mu_{A_k}(x) = \exp\left[-\left(\frac{x_{ji}-\tau_k}{2\sigma_k}\right)^2\right], \quad k = 1,2,3,4\,, i = 1,2, j = 1, \dots, n \qquad (13)$$

$o_{l,k}$ is the output of the k^{th} node of the layer l. In this paper, Gaussian membership function is considered. τ_k and σ_k parameters represent the center and the spread, respectively.

In the second layer, the nodes are also fixed. This layer is called as rule layer. In this layer, the outputs can be calculated as follows:

$$o_{2,k} = w_{jk} = \mu_{A_k}(x_{j1}).\, \mu_{A_k}(x_{j2}), k = 1,2,3,4, j = 1, \dots, n. \qquad (14)$$

In the third layer, the nodes are fixed nodes. This layer is called normalized firing strength. The outputs of this layer can be calculated as:

$$o_{3,k} = \overline{w}_{jk} = \frac{w_{jk}}{\sum_{k=1}^{4} w_{jk}}, k = 1,2,3,4, j = 1, \dots, n. \quad (15)$$

In the fourth layer, which is called defuzzification layer, the node is an adaptive one. The association node function in this layer is a linear one and the outputs can be represented as below:

$$o_{4,k} = \overline{w}_{jk} f_k = \overline{w}_{jk} p_i^k, \ k = 1,2,3,4, \ i = 0,1,2. \quad (16)$$

In this study, p_i^k will be assumed to be a symmetric triangular fuzzy number.

In the fifth layer, the single node carries out the sum of inputs of all the layers. It is called as the summation layer. The overall output of the structure is obtained as bellow:

$$o_{5,k} = \sum_{k=1}^{4} \overline{w}_{jk} f_k. \quad (17)$$

3. Methodology of the proposed method

In Eq. (17), assume $x_j = (1, x_{j1}, x_{j2}, \dots,, x_{jp})$ is a p-dimensional input vector of the independent variables at the j^{th} observation, also, $P = (p_0, p_1, \dots, p_p)$ is a vector of unknown fuzzy parameters, $Y_j = (a_{y_j}, \beta_{y_j})$ and $\widehat{Y}_j = \left(\hat{a}_{y_j}, \hat{\beta}_{y_j}\right)$ are the j^{th} observed value and estimated value of the dependent variables for , $j = 1, \dots, n$, where n is the number of data points, a_{y_j} is center value and β_{y_j} is spread value of Y_j, and $\hat{a}_{y_j}$ is center value and $\hat{\beta}_{y_j}$ is spread value of $\widehat{Y}_j$. p_i, $i = 0, \dots, p$, can be denoted in vector form as $p_i = \{b, \alpha\}$ where $b = (b_0^k, b_1^k, \dots, b_p^k)$ and $\alpha = (\alpha_0^k, \alpha_1^k, \dots, \alpha_p^k)$, $k = 1, \dots, m$, where b_i^k is center value and α_i^k is spread value of p_i, $i = 0, \dots, p$. So from the above definitions, using fuzzy arithmetic and substituting p_i^k into Eq. (17), it can be expressed as:

$$\widehat{Y}_j = \sum_{k=1}^{m} \sum_{i=0}^{p} b_i^k \overline{w}_{jk} x_{ji} + \sum_{k=1}^{m} \sum_{i=0}^{p} \alpha_i^k \overline{w}_{jk} x_{ji}. \quad (18)$$

So,

$$\hat{a}_{y_j} = \sum_{k=1}^{m} \sum_{i=0}^{p} b_i^k \overline{w}_{jk} x_{ji}, \text{ and } \hat{\beta}_{y_j} = \sum_{k=1}^{m} \sum_{i=0}^{p} \alpha_i^k \overline{w}_{jk} x_{ji}, \quad (19)$$

where $\overline{w}_k$ is known.

In the proposed methods, the premise and consequence parameters are obtained by method ANFIS for only the first part of the Eq. (18). Then, the obtained premise and consequence parameters are put in a matrix P=XP_0. $X \in [10^{-\gamma}, 10^{\gamma}]$. In the following, the obtained parameters are optimized by meta-heuristic algorithms such as PSO and ACO_R. In the finally, the consequence parameters are computed for center and spread value Y_j as follows:

Step 1: The optimization premise parameters are put in Eq. (18).

Step 2: The consequence parameters b_i^k and α_i^k are obtained by solving the following linear programming (LP) model:

$\min \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{i=0}^{p} \alpha_i^k \bar{w}_k x_{ji}$,

So that, the following constraints must be established:

$$\sum_{k=1}^{m} \sum_{i=0}^{p} b_i^k \bar{w}_k x_{ji} - (1-h) \sum_{k=1}^{m} \sum_{i=0}^{p} \alpha_i^k \bar{w}_k x_{ji} \le a_{y_j} - (1-h)\, \beta_{y_j},$$

$$\sum_{k=1}^{m} \sum_{i=0}^{p} b_i^k \bar{w}_k x_{ji} + (1-h) \sum_{k=1}^{m} \sum_{i=0}^{p} \alpha_i^k \bar{w}_k x_{ji} \ge a_{y_j} + (1-h)\, \beta_{y_j}\,, \qquad (20)$$

and $\sum_{k=1}^{m} \sum_{i=0}^{p} \alpha_i^k \bar{w}_k \ge 0, \;\; i = 0, \dots, p, \quad k = 1, \dots, m, \;\; j = 1, \dots, n.$

In this paper, we show the proposed methods methodology in diagram (1) and use the MATLAB software tool for codding.

To evaluate the performance of the various methods, the error rate can be calculated using Eq. (2) as below:

$$\text{ERROR}=\frac{1}{n}\sum_{j=1}^{n}(Y_j - \hat{Y}_j)^2=\frac{1}{n}\sum_{j=1}^{n}\left(3\left(a_{y_j} - \sum_{k=1}^{m}\sum_{i=0}^{p} b_i^k \bar{w}_{jk} x_{ji}\right)^2 + 2\left(\beta_{y_j} - \sum_{k=1}^{m}\sum_{i=0}^{p} \alpha_i^k \bar{w}_{jk} x_{ji}\right)^2\right). \quad (21)$$

Eq. (21) is used as a quantity to measure bias between the observed values, $Y_j = (l_j, a_j, r_j)$, and the predicted values, $\hat{Y}_j = (\hat{l}_j, \hat{a}_j, \hat{r}_j)$, for all X_js $(j = 1, \dots, n)$ where l_j, a_j , r_j , $\hat{l}_j, \hat{a}_j$, and $\hat{r}_j$ are lower, center, and upper of the observed fuzzy outputs and lower, center, and upper of the estimated fuzzy outputs.

4. Simulation and Numerical examples

In order to illustrate the applicability of the proposed algorithms, the following simulation and practical examples are considered. Also, the obtained results of the different methods are compared.

Example 1: Consider the following functions:

$f(x_1, x_2) = 24.23r^2(0.75 - r^2) + 5,$

$r^2 = (x_1/10 - 0.5)^2 + (x_2/10 - 0.5)^2.$

Suppose the domain of $X = (x_1, x_2)$ is $D = [0,10]^2$. A set of data is generated the same way as in (Dastranj 2011) that summarized in the following manner.

The independent variables x_1 and x_2 are crisp inputs. They are randomly taken from 0 to 10. Let output $Y_j = (a_j, \beta_j)$ (j $= 1,2, \dots, n)$ is a symmetric fuzzy number and so that

$$\begin{cases} a_j = f(x_{j1}, x_{j2}), \\ \beta_j = (1/4) f(x_j) + rand\ [0,1], \end{cases} j = 1, \dots, 30,$$

Where the rand $[a, b]$ denotes a random number between a and b for each j. For fitting regression model, we apply the different methods. Also, to evaluate the performance of the different methods, we use the error value ERROR numerically.

1. Divide all data into two subsets, test dataset (20% of dataset) and train dataset (80% of rest dataset). 2. Determine the initial values of the premise and the consequence parameters by ANFIS method for only centers. 3. Construct candidate solutions in a probabilistic way using a probability distribution over the search space. 4. Use the candidate solutions for modification the probability distribution in a way. 5. Repeat for variable n until a solution is made for all variables and N ants. 6. Sort all of solution archive, save k better answer and clear the rest of the answers. 7. Get premise parameters.	8. Put in Eq. (18) and estimate consequence parameters for centers and spread by linear programming and Eq. (20). 9. Determine the initial values of positions and velocities for all particles. 10. Update positions according to Eq. (7). 11. Compute and evaluate the objective function at each particle position. 12. Compute $x^{l,tseb}[t]$ and $x^{g,tseb}[t]$ for $j = 1, \dots, n$. 13. Update velocity according to Eq. (6). 14. The lack of progress, reach to the predetermined repetitions number or converge to a certain amount. 15. Calculate ERROR. 16. Evaluate the chosen net by using the test dataset. 17. Report errors of the different methods and results of net in test and train dataset.

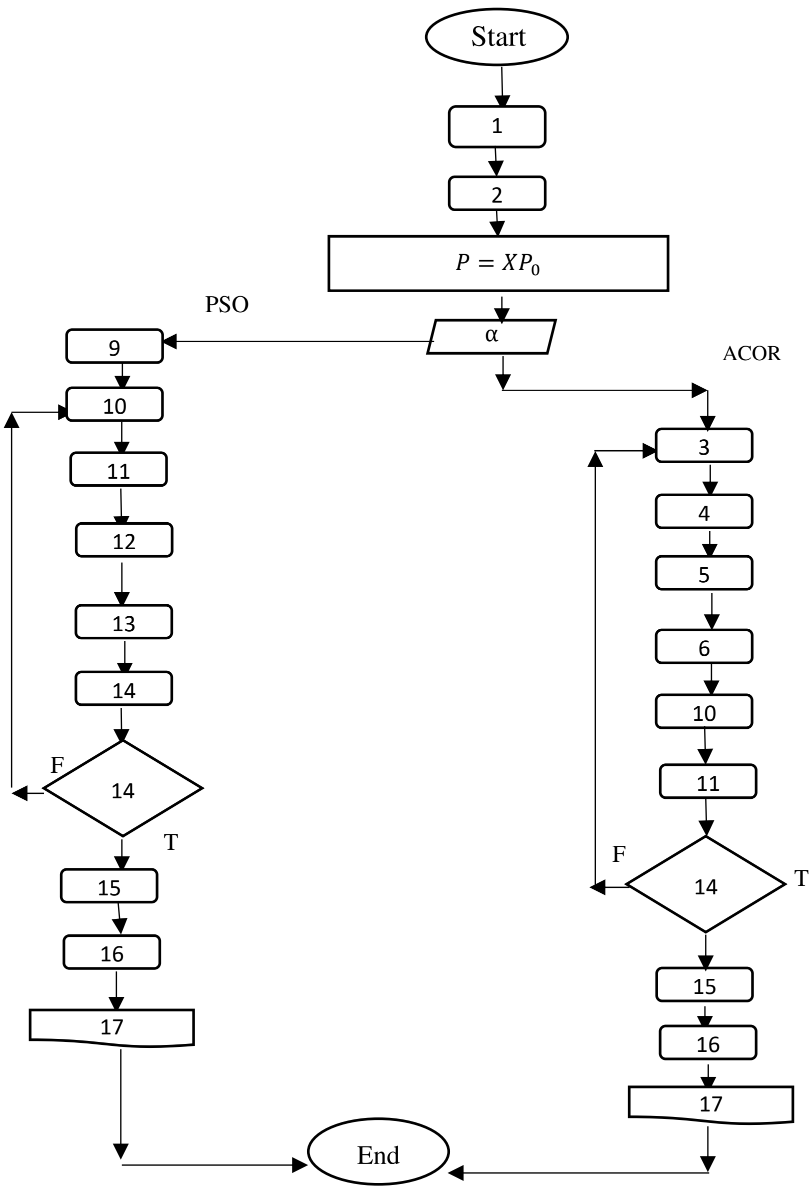


Fig 1: Diagram of the proposed methods methodology

In the following, the obtained parameters of using the FWLP method and the proposed methods are respectively displayed in Tables 2-5. Also, the obtained regression model in LP method is as follows:

$$\hat{Y}_j = \left(\hat{a}_j, \hat{\beta}_j\right) = (0.0744,2.7036) + (1.4498,0.2699)X_{j1} + (0.4007,0.1173)X_{j2}.$$

In Tables 5 and 6, the obtained results of the different methods are summarized. It can be observed that the error values of the proposed methods are lower than the error values the other ones. Thus, the proposed algorithms provide the better prediction than the LP and FWLP methods.

Tables 2: The obtained premise and consequence parameters by using the FWLP method.

Variable	k	(τ_k, σ_k)	(b_0^k, α_0^k)	(b_1^k, α_1^k)	(b_2^k, α_2^k)
x_1	1	(4.5057,1.5576)	(5.3794,0.7065)	(1.6089,0.4154)	(-0.2687,0.2711)
	2	(8.3810,2.3388)	(-1.5851,0.7099)	(1.8465, 0.5218)	(0.6591,0.0417)
x_2	3	(6.4446,2.8907)	(1.0780,0.5292)	(1.6831,0.4525)	(0.0664,0.0352)
	4	(3.6626,1.9964)	(-24.3688,0.8157)	(4.9530, 0.2096)	(-0.819, 0.2715)

Tables 3: The obtained premise and consequence parameters by using the FWPSOLP method.

Variable	k	(τ_k, σ_k)	(b_0^k, α_0^k)	(b_1^k, α_1^k)	(b_2^k, α_2^k)
x_1	1	(2.5841,0.6631)	(7.9135,0.8235)	(-3.0217,0.3408)	(2.0413,0.2575)
	2	(0.9110,4.2916)	(-6.0916, 0.9564)	(4.8651,0.3058)	(-0.2928,0.1275)
x_2	3	(5.6721,0.9188)	(-2.7667,0.4868)	(1.3454,0.5150)	(0.8346,0.1790)
	4	(7.5296,1.9631)	(-0.4816,1.5689)	(1.5028,0.4235)	(0.4590,0.0138)

Tables 4: The obtained premise and consequence parameters by using the FWACORLP method.

Variable	k	(τ_k, σ_k)	(b_0^k, α_0^k)	(b_1^k, α_1^k)	(b_2^k, α_2^k)
x_1	1	(-11.8349,13.0272)	(-0.7995,1.6825)	(1.5435,0.4268)	(0.4729,0.0042)
	2	(2.7250,0.5315)	(3.4915,0.2382)	(1.3089, 0.5484)	(-0.1406,0.0894)
x_2	3	(16.3264,23.0656)	(-6.1997,0.5411)	(4.7596, 0.2361)	(-0.2099,0.1572)
	4	(5.6384,0.8497)	(6.6085,0.4976)	(-8.5886,0.2493)	(4.6917,0.6898)

Table 5. The predicted fuzzy outputs using the different methods for Example 1.

K	x_1	x_2	$Y_j = (a_j, \beta_j)$	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ LP	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ FWLP	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ FWPSOLP	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ FWACORLP
1	6.6280	6.6090	(11.6800,3.6280)	(12.3319,5.2680)	(11.6917,4.2428)	(12.1748,4.7132)	(12.0779,4.5117)
2	9.2970	7.1810	(16.9620,4.8400)	(16.4306,6.0555)	(16.9672,4.9535)	(16.5664,5.8050)	(16.5508,5.7319)
3	3.1780	7.1450	(7.7340,2.4680)	(7.5450,4.3997)	(8.0370,3.5867)	(7.9698,3.0730)	(7.9565,3.0590)
4	1.1940	6.2100	(3.9620,1.2360)	(4.2940,3.7545)	(4.5495,2.4290)	(4.4525,2.2171)	(4.2483,1.8955)
5	3.9220	8.9840	(9.7070,3.1400)	(9.3606,4.8162)	(9.0832,4.4044)	(9.6557,3.3483)	(9.6545,3.3214)
6	2.6390	5.7270	(7.2780,1.8480)	(6.1953,4.0878)	(6.7819,2.9067)	(7.4680,2.8757)	(7.3985,3.2506)
7	9.7360	8.8430	(17.8320,4.4820)	(17.7331,6.3690)	(17.9882,5.2545)	(18.2069,5.8188)	(18.4069,5.8752)
8	3.7110	3.8310	(6.9900,1.9900)	(6.9897,4.1547)	(7.2932,2.9339)	(6.6338,3.0668)	(7.3557,2.9512)
9	2.8300	3.6310	(5.8000,2.2370)	(5.6323,3.8935)	(5.8992,2.5099)	(5.8110,2.5897)	(5.8015,2.2402)
10	7.1880	1.7980	(11.4750,3.3530)	(11.2159,4.8547)	(11.4573,3.3884)	(11.1191,4.6369)	(11.1456,4.7579)
11	6.3490	2.3940	(9.5760,3.2770)	(10.2383,4.6982)	(9.5565,3.4622)	(10.0389,4.2857)	(10.1334,4.4020)
12	1.3000	4.8520	(4.1170,1.7250)	(3.9034,3.6237)	(4.0000,2.0636)	(3.9500,2.1155)	(4.0699,1.9226)
13	0.5160	5.9760	(3.2490,1.0740)	(3.2172,3.5440)	(3.1623,2.0436)	(3.0080,1.8463)	(3.0801,1.4873)
14	7.5790	3.5590	(12.3730,3.2320)	(12.4884,5.1669)	(12.1927,3.7952)	(11.7823,4.8986)	(12.6000,4.9201)
15	1.0710	7.4820	(4.4630,1.7000)	(4.6254,3.8705)	(4.6572,2.8348)	(4.2769,2.1708)	(4.2929,2.1109)
16	7.1060	2.0840	(11.2450,2.9840)	(11.2116,4.8661)	(11.1546,3.4413)	(11.0951,4.6054)	(11.1544,4.7241)
17	5.2480	6.6670	(10.1170,3.2140)	(10.3545,4.9023)	(10.4973,4.0892)	(10.2359,4.0733)	(10.0989,3.8683)
18	2.9670	8.8280	(6.7840,2.2730)	(7.9135,4.5402)	(7.6386,4.0518)	(6.9426,2.9709)	(6.7717,2.7632)
19	3.6090	8.8930	(8.9790,2.8670)	(8.8703,4.7211)	(8.6220,4.2868)	(8.9859,3.2155)	(9.0228,3.1034)
20	3.9190	8.9900	(9.3860,2.5520)	(9.3586,4.8161)	(9.0778,4.4053)	(9.6527,3.3471)	(9.6524,3.3196)
21	5.6960	9.6320	(12.0780,3.0950)	(12.1921,5.3711)	(11.6004,4.7585)	(12.4997,4.1147)	(12.5471,4.1537)
22	3.1610	7.1270	(8.0850,2.6180)	(7.5131,4.3930)	(8.0091,3.5741)	(7.9441,3.0679)	(7.9362,3.0629)

23	5.8020	0.2170	(8.9130,2.8190)	(8.5729,4.2951)	(8.8643,3.0698)	(8.3372,4.0292)	(8.2586,4.1598)
24	0.8070	9.8370	(4.8600,1.5760)	(5.1863,4.0755)	(3.9515,3.4784)	(4.9704,2.0578)	(5.0752,2.0683)
Data Test							
1	6.3490	2.6600	(10.1150,2.8770)	(10.3449,4.7294)	(9.5709,3.5088)	(10.0718,4.2888)	(10.2611,4.4023)
2	6.3490	2.3940	(9.5760,3.2770)	(10.2383,4.6982)	(9.5565,3.4621)	(10.0391,4.2854)	(10.1334,4.4019)
3	5.2480	6.6670	(10.1170,3.2140)	(10.3545,4.9023)	(10.4972,4.0890)	(10.2360,4.0731)	(10.0990,3.8684)
4	2.9670	8.8280	(6.7840,2.2730)	(7.9135,4.5402)	(7.6383,4.0517)	(6.9428,2.9707)	(6.7715,2.7631)
5	3.9190	8.9900	(9.3860,2.5520)	(9.3586,4.8161)	(9.0775,4.4051)	(9.6528,3.3468)	(9.6525,3.3198)
6	5.6960	9.6320	(12.0780,3.0950)	(12.1921,5.3711))	(11.6002,4.7583)	(12.4997,4.1142)	(12.5472,4.1540)

Tables 6: The obtained error results by using the different methods for Example 1.

Different methods	Value ERROR of train	Value ERROR of test
LP	8.2812	8.8349
FWLP	2.6819	4.2682
FWPSOLP	2.1084	2.2208
FWACORLP	2.0665	2.3023

Example 2: Application of the Proposed Method to Predict Surface Roughness in Machining GFRP Composites

Surface roughness is an important factor in manufacturing engineering because a rough surface has negative effects on the fatigue strength, creep life, wear and corrosion resistance of various components. In turning operation, high quality turned surface eliminates the need for secondary operations. Under ideal cutting conditions, the texture of machined surface comprises regular traces of the tool nose (Fig. 2). The surface geometry of a workpiece depends upon a number of factors including machining variables, tool geometry, materials of the workpiece and tool, and the vibrations of the machine tool system (Boothroyd 2011, Danesh 2015). The deviations may be repetitive or random and may result from roughness, waviness, lay and flaws (Lu 2007).

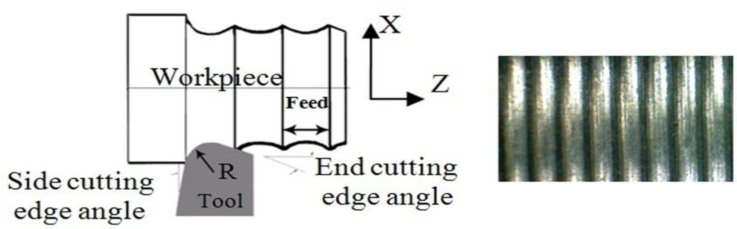
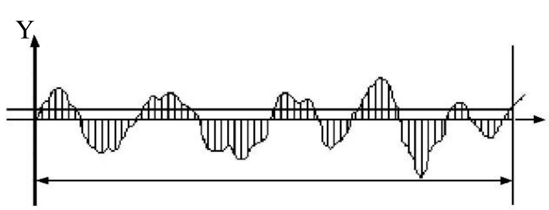


Fig. 2. Formation of feed marks in the surface of the turned part (Danesh 2015).

There are many different roughness parameters in use, but the arithmetical mean roughness (Ra) value is the most widely used parameter for the surface roughness assessment. Ra can be calculated using the following formula:

$$R_a = \frac{\left|\sum A\right|}{L}$$

Where A is the area between the roughness profile and its central line and L is the sampling length. Fig. 2 shows how *R*a is defined.



A X

L

Fig.2. The basic characteristic of the surface texture is its irregularity and randomness.

The input process parameters are cutting speed (V) in m/min, feed (f) in mm/rev, depth of cut (d) in mm, and fiber orientation angle of the work piece (ø), in degrees. The process parameter values and observed response, i.e., surface roughness (Ra) values for all the 31 experiments are listed in Table 14 (Sarma 2007).

In the following, the obtained premise parameters the FWLP method and the proposed methods have been shown in Table 7. For fitting regression model, the different methods are applied and the obtained outputs are summarized in Table 8. The number of the consequence parameters is 160. Thus, the results are not reported here in consideration of the limited space. The obtained regression model using the LP method is shown as follows:

$$\hat{Y}_j = (\hat{a}_j, \hat{\beta}_j) = (1.5820,0.1906) + (-0.0061,0.0011)X_{j1} + (9.3822,0.3185)X_{j2} + (0.0783,0.1110)X_{j3} + (0.0068,0.0004)X_{j4}.$$

In order to evaluate the different methods, value ERROR in equation 21 is calculated and reported in Table 9. Like the previous examples based on the error values, we can be observed that the proposed algorithms have lower error than the LP and FWLP methods in the prediction.

Table 7: The obtained premise parameters by using the FWLP method and the proposed methods for Example 2.

	(τ_{ki},σ_{ki})	FWLP	FWPSOLP	FWACORLP
x_1	(τ_{11},σ_{11})	(190.9591,37.2341)	(115.8919,105.2195)	(83.7398,49.3367)
	(τ_{12},σ_{12})	(101.6400,49.3308)	(192.3916,74.8500)	(83.7398,49.3367)
x_2	(τ_{21},σ_{21})	(0.2690,0.0942)	(0.1512,0.0255)	(83.7398,49.3367)
	(τ_{22},σ_{22})	(0.3458,0.0909)	(0.0842,0.0906)	(-0.2910,0.0882)
x_3	(τ_{31},σ_{31})	(0.8215,0.5680)	(-0.6583,0.9815)	(-1.4056,0.5670)
	(τ_{32},σ_{32})	(0.8208,0.2302)	(-0.6245,0.5186)	(1.1731,0.2753)
x_4	(τ_{41},σ_{41})	(70.1730,18.9116)	(68.6146,9.7259)	(-148.3552,30.0098)
	(τ_{42},σ_{42})	(59.8715,20.0489)	(43.1687,19.2976)	(86.5847,8.0516)

Table 8. The predicted fuzzy outputs using the different methods for Example 2.

K	$V\ m/min$ (x_1)	$f\ mm/rev$ (x_2)	$d\ mm$ (x_3)	$\emptyset$ (x_4)	surface roughness, $Ra\ \mu m$ $Y_j = (a_j, \beta_j)$	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ LP method	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ FWLP method	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ FWPSOLP method	$\hat{f}(x_j) = (\hat{a}_j, \hat{\beta}_j)$ FWACORLP method
1	82	0.0960	0.5000	45	(2.5430,0.0890)	(2.3285,0.3853)	(2.5436,0.1850)	(2.5640,0.1702)	(2.4494,0.4009)
2	194	0.0960	0.5000	45	(2.3110,0.1460)	(1.6451,0.5070)	(2.3109,0.1580)	(2.3643,0.3417)	(2.3110,0.1530)
3	82	0.1910	0.5000	45	(3.1475,0.3935)	(3.2198,0.4155)	(3.1478,0.3967)	(3.1477,0.3972)	(3.1673,0.4356)
4	82	0.0960	1.0000	45	(2.3060,0.1170)	(2.3677,0.4408)	(2.3057,0.2570)	(2.3063,0.1247)	(2.5045,0.5190)
5	82	0.1910	1.0000	45	(3.1535,0.1515)	(3.2590,0.4710)	(3.1553,0.5509)	(3.1530,0.1588)	(3.2249,0.5309)
6	82	0.0960	0.5000	75	(2.5350,0.0710)	(2.5332,0.3983)	(2.5355,0.1372)	(2.5527,0.1446)	(2.5928,0.4044)
7	194	0.0960	0.5000	75	(2.4355,0.0965)	(1.8498,0.5200)	(2.4342,0.1352)	(2.4842,0.2942)	(2.4426,0.1550)
8	82	0.1910	0.5000	75	(3.3920,0.1970)	(3.4245,0.4285)	(3.3928,0.2859)	(3.3490,0.3561)	(3.3009,0.4218)
9	82	0.0960	1.0000	75	(2.3810,0.0540)	(2.5724,0.4538)	(2.3808,0.1905)	(2.3854,0.1026)	(2.6149,0.5236)
10	194	0.0960	1.0000	75	(2.2935,0.2025)	(1.8889,0.5755)	(2.2927,0.2051)	(2.2944,0.2090)	(2.2940,0.2072)
11	194	0.0960	1.0000	75	(2.2935,0.2025)	(1.8889,0.5755)	(2.2927,0.2051)	(2.2944,0.2090)	(2.2940,0.2072)
12	194	0.0960	1.0000	75	(2.2935,0.2025)	(1.8889,0.5755)	(2.2927,0.2051)	(2.2944,0.2090)	(2.2940,0.2072)

13	82	0.1910	1.0000	75	(3.1140,0.1050)	(3.4637,0.4840)	(3.1123,0.3966)	(3.1260,0.1539)	(3.3284,0.5347)
14	194	0.1910	1.0000	75	(3.1205,0.1355)	(2.7802,0.6057)	(3.1194,0.2981)	(3.0485,0.3170)	(3.1002,0.2090)
15	54	0.1430	0.7500	60	(3.0680,0.2070)	(3.0623,0.4041)	(3.0687,0.2564)	(3.0670,0.2167)	(2.9703,0.4036)
16	126	0.0480	0.7500	60	(2.1505,0.1355)	(1.7316,0.4521)	(2.1506,0.2170)	(2.1496,0.1398)	(1.9954,0.4535)
17	126	0.2380	0.7500	60	(3.3760,0.1810)	(3.5142,0.5126)	(3.3247,0.8904)	(3.3769,0.1859)	(3.4062,0.3666)
18	126	0.1430	1.2500	60	(2.9440,0.0190)	(2.6621,0.5378)	(2.9449,0.2051)	(2.8996,0.1546)	(2.7466,0.5454)
19	126	0.1430	0.7500	30	(2.5355,0.1905)	(2.4182,0.4693)	(2.5361,0.6216)	(2.5408,0.3139)	(2.5662,0.2502)
20	126	0.1430	0.7500	90	(2.9095,0.1515)	(2.8276,0.4953)	(2.9097,0.3441)	(2.8818,0.5248)	(2.8337,0.4663)
21	126	0.1430	0.7500	60	(2.9480,0.0480)	(2.6229,0.4823)	(2.7413,0.4671)	(2.7414,0.4626)	(2.7414,0.4624)
22	126	0.1430	0.7500	60	(2.7520,0.2110)	(2.6229,0.4823)	(2.7413,0.4671)	(2.7414,0.4626)	(2.7414,0.4624)
23	126	0.1430	0.7500	60	(2.7980,0.1910)	(2.6229,0.4823)	(2.7413,0.4671)	(2.7414,0.4626)	(2.7414,0.4624)
24	126	0.1430	0.7500	60	(2.5940,0.1660)	(2.6229,0.4823)	(2.7413,0.4671)	(2.7414,0.4626)	(2.7414,0.4624)
25	126	0.1430	0.7500	60	(2.7105,0.1845)	(2.6229,0.4823)	(2.7413,0.4671)	(2.7414,0.4626)	(2.7414,0.4624)
Test Data									
1	194	0.1910	0.5000	75	(3.1165,0.1345)	(2.7411,0.5503)	(3.3085,0.2054)	(3.2415,0.7498)	(3.2796,0.1654)
2	194	0.0960	1.0000	75	(2.2935,0.2025)	(1.8889,0.5755)	(2.2927,0.2051)	(2.2944,0.2090)	(2.2940,0.2072)
3	302	0.1430	0.7500	60	(2.3455,0.3495)	(1.5489,0.6736)	(1.7177,0.2243)	(2.5398,0.7184)	(3.1909,0.1059)
4	126	0.1430	0.2500	60	(2.8360,0.1180)	(2.5837,0.4268)	(3.7428,0.0932)	(2.8193,0.9863)	(2.6376,0.1792)
5	126	0.1430	0.7500	60	(2.6360,0.1930)	(2.6229,0.4823)	(2.7413,0.4671)	(2.7414,0.4626)	(2.7414,0.4624)
6	126	0.1430	0.7500	60	(2.6800,0.1960)	(2.6229,0.4823)	(2.7413,0.4671)	(2.7414,0.4626)	(2.7414,0.4624)

Tables 9: The obtained error results by using the different methods for Example 2.

Different methods	Value ERROR of train	Value ERROR of test
LP	0.4857	0.7291
FWLP	0.0127	0.6907
FWPSOLP	0.0585	0.5051
FWACORLP	0.1139	0.4669

5. Conclusion

The aim of this study is to develop ANFIS method by PSO and ACO_R algorithms, and the linear programming. In this paper, we considered the fuzzy regression model with crisp inputs and triangular fuzzy output. So, the proposed algorithms are fuzzified to fit the fuzzy regression function with the linear programming. The effectiveness of various methods is demonstrated by different examples. By considering the obtained results of different methods and figures, it can be determined that the performance of the proposed methods is evidently better than the ANFIS method and linear programming. Based on Examples, it should be pointed out that the performance of the prediction can be significantly enhanced by using the proposed algorithms in case two-dimensional input. Some conclusions are summarized as follows:

1) A comparison with ANFIS and linear programming methods verifies high computational precision and efficiency of the proposed methods. The results demonstrate that proposed methods are efficient methods for the in many practical engineering problems with uncertainty.

2) By using the obtained results and tables, it can be determined that the proposed methods are feasible. Based on examples, it should be pointed out that the proposed methods have more accurate than the LP and FWLP methods. Also, they decrease the error values to a minimum level.
3) In the proposed methods, the used constrains assure that the support of the estimated values from the regression model includes the support of the observed values in h-level$(0 < \mathrm{h} \leq 1)$. In linear programming, the width of the estimated value depends on the observation number. As the number increases, there will be a rise in the width of the estimated value which can, in turn, be weakened by the proposed methods.
4) In addition, the proposed methods have not more complicated than the FWLP method in computations but they have more accurate than it.

5) The obtained results from the simulated example and the case study in the field of prediction of surface roughness in machining GFRP composites show that the presented methods are especially

useful for practical problems which involve output parameters that include some degree of uncertainty, inhomogeneity, randomness and imprecision in the output data.

6) The proposed approach can enrich the methods that offer a promising way for further research on the design and optimization of fuzzy neural networks.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This research does not involve human nor animals.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Boothroyd G., Knight W. A. (2006) Fundamentals of machining and machine tools Taylor and Francis: 166-173

Danesh M., Khalili K. (2015) Determination of Tool Wear in Turning Process Using Undecimated Wavelet Transform and Textural Features. Procedia Technology 19: 98-105

Danesh S., Farnoosh R., Razzaghnia T. (2016) Fuzzy nonparametric regression based on adaptive neuro-fuzzy inference system, Neurocomputing 173 : 1450-1460,.

Danesh M., Danesh S., Khalili K. (2020) Multi-Sensory Data Fusion System for Tool Condition Monitoring Using Optimized Artificial Fuzzy Inference System, Aerospace Mechanics Journal 15(2): 103-118 (in Persian)

Danesh M. and Danesh S. (2020) A Combinatorial Algorithm for Fuzzy Parameter Estimation with Application to Uncertain Measurements, Journal of AI and Data Mining 8 (4), No 4: 525-533.

Danesh S., Fuzzy Parameters Estimation via Hybrid Methods (2018) Hacettepe Journal of Mathematics and Statistics 47 (6): 1605 - 1624.

Diamond P., Fuzzy least squares, Information Sciences 46(3), 1988, 141-157

Dastranj MR., Ebrahimi E., Changizi N., Sameni E. (2011) Control DC Motorspeed with Adaptive Neuro-Fuzzy control.Australian Journal of Basic and Applied Sciences 5,:1499-1504.

Eberhart R., Kennedy J. (1995) Particle swarm optimization. International Conference of Neural Networks, Perth, Australia

Faes, M., Moens, D. Recent (2020) Trends in the Modeling and Quantification of Non-probabilistic Uncertainty. Arch Computat Methods Eng 27: 633–671.

Fan Y.R., Huang G.H., Yang A.L. (2013) Generalized fuzzy linear programming for decision making under uncertainty. Feasibility of fuzzy solutions and solving approach, Information Sciences 241: 12-27.

Hanss M. Applied fuzzy arithmetic (2005) an introduction with engineering applications. Springer, Berlin

Hidalgo D., Melin P., Castillo O. (2012) An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms, Expert Systems with Applications 39 (4): 4590-4598.

Hemayatkar N., Khalili-Damghani K., Didehkhani H. et al. (2019) Developing a fuzzy inference system to devise proper business strategies: a study on carpet industry. J Ind Eng Int 15: 529–544

Hong D.H., Song J. K, Young H. (2001) Fuzzy least-squares linear regression analysis using shape preserving operations. Information Sciences 138: 185-193.

H. Ishibushi, H. Tanaka, H. Okado, An architecture of neural networks with interval weights and its application to fuzzy regression analysis, Fuzzy Sets and Systems, 57(1) , pp 27-39.

Isukapalli S. S., Roy A., and Georgopoulos P. G. (1998) Stochastic Response Surface Methods (SRSMs) for Uncertainty Propagation: Application to Environmental and Biological Systems, Risk Analysis 18(3): 351-363.

Jang J.S.R. (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cyber 23(3): pp 665-685.

Khalili K., Danesh M. (2015) Identification of vibration level in metal cutting using undecimated wavelet transform and gray-level co-occurrence matrix texture features, 229(2):786–804.

Kumar, A., Kaur, J. & Singh P.(2011) A new method for solving fully fuzzy linear programming problems. Applied Mathematical Modelling 35(2): 817-823.

Lu C. (2007) Study on prediction of surface quality in machining process. Journal of materials processing technology 24: 439-450.

Razzaghnia. T, Danesh S., Maleki A. (2011) Hybrid fuzzy regression with trapezoidal fuzzy data, Proc. SPIE 8349,834921

Razzaghnia T., Danesh S. (2015) Nonpzarametric Regression with Trapezoidal Fuzzy Data, International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC) , 3(6): 3826 – 3831.

Sarma P.M.M.S., karunamoothy L., Palanikumar K. (2009) Surface roughness parameters evaluation in machining GFRP Composites by PCD tool using digital image processing, Journal of Reinforced Plastics and Composites, 28: 1567-1585

Socha K., Dorigo M. (2008) Ant colony optimization for continuous domains. Eur. J. Oper. Res. 185: 1155–1173.

Tanaka, H., Uejimas, S., Asia, K. (1982) Linear regression analysis with fuzzy model. IEEE Transactions on Systems, Man and Cybernetics 12:903-907

Tanaka H. (1987) Fuzzy data analysis by possibilistic linear models, Fuzzy Sets and Systems 24(3): 363-375

Tanaka H., Hayashi I., Watada J. (1989) possibilistic linear regression analysis for fuzzy data, European Journal of Operational Research, 40 (3): 389-396

Tanaka H., Uejima S., Asia K. (1982) Linear regression analysis with fuzzy model, IEEE Transactions on Systems, Man, and Cybernetics 12: 903-907.

Tanaka H., Watada J. (1988) Possibilistic linear systems and their application to the linear regression model, Fuzzy Sets and Systems 27 (3): 275-289

Tanaka H. (1987) Fuzzy data analysis by possibilistic linear models. Fuzzy Sets and Systems 24(3): 363-375.

Wang, C. Matthies, H.G. (2019) Epistemic uncertainty-based reliability analysis for engineering system with hybrid evidence and fuzzy variables. Comput. Methods Appl. Mech. Eng. 355: 438–455.

Wang C., Matthies H. G., Coupled fuzzy-interval model and method for structural response analysis with non-probabilistic hybrid uncertainties, Fuzzy Sets and Systems, 2020,

Wang P., Zhu H., Tian H., Cai G. (2021) Analytic target cascading with fuzzy uncertainties based on global sensitivity analysis for overall design of launch vehicle powered by hybrid rocket motor, Aerospace Science and Technology 106680

Y. Wang, L. Li, K. Wang, An online operating performance evaluation approach using probabilistic fuzzy theory for chemical processes with uncertainties, Computers & Chemical Engineering, Volume 144, 2021, 107156.

Zhang, C. & Guo, P. (2018) FLFP: A fuzzy linear fractional programming approach with double-sided fuzziness for optimal irrigation water allocation.Agricultural Water management, vol. 199: 105–119.

Zhang X., Lu Z., Wang L., Li G. (2020) The importance measure of fuzzy input on failure credibility under the fuzzy uncertainty, Aerospace Science and Technology 107: 106320.