A Tabu–Harmony Search-Based Approach to Fuzzy Linear Regression

M. Hadi Mashinchi, Mehmet A. Orgun, *Senior Member, IEEE*, Mashaallah Mashinchi, and Witold Pedrycz, *Fellow, IEEE*

***Abstract*—We propose an unconstrained global continuous optimization method based on tabu search and harmony search to support the design of fuzzy linear regression (FLR) models. Tabu and harmony search strategies are used for diversification and intensification of FLR, respectively. The proposed approach offers the flexibility to use any kind of an objective function based on client's requirements or requests and the nature of the dataset and then attains its minimum error. Moreover, we elaborate on the error produced by this method and compare it with the errors resulting from the other known estimation methods. To study the performance of the method, three categories of datasets are considered: Numeric inputs–symmetric fuzzy outputs, symmetric fuzzy inputs–symmetric fuzzy outputs, and numeric inputs–asymmetric fuzzy outputs. Through a series of experiments, we demonstrate that in terms of the produced error with different model-fitting measurements, the proposed method outperforms or is Pareto-equivalent to the existing methods reported in the literature.**

***Index Terms*—Fuzzy linear regression (FLR), fuzzy modeling, global continuous optimization, harmony search, imprecise relationship, tabu search.**

I. Introduction

CLASSICAL regression analysis offers a conceptual and algorithmic vehicle to discover relationships (functions) between independent (explanatory, covariant, input) variables and dependent (response, output, model's estimated output) variables [1], [2]. The problem is to optimize the given function's parameters for the given input–output data so that a predetermined objective function (error) attains its (global) minimum [3]. The ultimate objective of forming a regression model is to estimate the value of a continuous dependent variable for any arbitrary value of the independent variable [4]. Regression models are important tools in operations research, complex systems analysis, and various fields of application such as economy, finance, marketing, social sciences, healthcare, and others [5], [6].

Manuscript received March 13, 2010; revised September 29, 2010; accepted November 15, 2010. Date of publication January 17, 2011; date of current version June 6, 2011. M. Machinshi was supported in part by the Fuzzy Systems and its Applications Center of Excellence, Shahid Bahonar University of Kerman, Kerman, Iran.

M. H. Mashinchi and M. A. Orgun are with the Department of Computing, Macquarie University, Sydney, N.S.W. 2109, Australia (e-mail: h_mashinchi@yahoo.com; mehmet.orgun@mq.edu.au).

M. Mashinchi is with the Department of Statistics, Faculty of Mathematics and Computer Science, Shahid Bahonar University of Kerman 76169-1411, Kerman, Iran (e-mail: mashinchi@mail.uk.ac.ir).

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6R 2G7, Canada, and also with the Systems Research Institute, Polish Academy of Sciences, Warsaw, 00-901 Poland (e-mail: pedrycz@ee.ualberta.ca).

Digital Object Identifier 10.1109/TFUZZ.2011.2106791

Classical regression realized by means of statistical techniques is applied successfully to analyze quantitative data and homogeneous observations [2]. The deviation between the observed data and the estimated data encountered in classical regression is because of the measurement error or random variations of parameters [7]–[9]. Such random variations can be represented as a normal distribution of some variance and zero mean, which makes statistical techniques effective in determining the functional relationship for such types of data [7].

However, often a probabilistic representation of data is not available or not suitable, and there are significant deviations between the observed data and the corresponding estimates because of the imprecision introduced by human judgement or the indefiniteness/vagueness of the model's structure [10], [11]. In such systems, uncertainty arises not because of randomness but because of the phenomenon of fuzziness [9]. In "conventional" approaches, a numeric form of data is considered to construct the model, even if the data come with some imprecision or uncertainty. As a result, this makes the estimated model not fully efficient since by considering data instead of uncertain data, some important information may have been overlooked or neglected [4].

The factor of uncertainty emerges in the system's behavior because of several reasons:

1) the high complexity of the environment, which necessitates the adaptation of abstraction (granulation of information) for generalization purposes [12];
2) the influence of human subjective judgement in the decision process or the involvement of human-machine interactions [10], [11], [13]; and
3) partially available information [14], because of misrecording or inaccurate measurements [15].

The design of a regression model in such environments has been a challenge to the classical regression approaches. In classical approaches, fuzzy data used is treated as ordinal data so that classical statistical approaches can be applied [16]. As mentioned earlier, there is often useful information that can be overlooked at the defuzzification stage. An intuitively appealing approach toward the fuzziness of a system is not to defuzzify the data but to take the vagueness into consideration in the level of inferencing [16]. The defuzzification of data is only applied at the decision stage when it is deemed to become necessary [16].

The representation of an experimental environment, which is governed by uncertainty or impreciseness, can involve interval or fuzzy data [17]. To derive the corresponding models in such environments, fuzzy regression (FR) or interval regression (which is regarded as a simple version of FR [18]) is considered.

The FR model is referred to as a fuzzy or possibilistic model of classical regression, while classical regression is based on the principles of statistics [7]. It has been observed that the FR model is more effective than the classical regression when the normality of error terms and dependent variables and the availability of a sufficiently large dataset (complete data conditions) are not satisfied [11], [19], [20]. In such cases, the FR model may be used to explore the imprecise relationship between dependent and independent variables for the given system. The relationship is estimated by minimizing some error criteria capturing various facets of uncertainty [8], [21]. There have been many diverse applications of fuzzy linear regression (FLR), such as research and development project evaluation [22], housing [11], [12], insurance [23], and many others.

In the FLR model, we are interested in finding a fuzzy function $\widetilde{y}$ in the form (1) that fits a finite number of numeric input-fuzzy output data $\left((x_{1i}, x_{2i}, \ldots, x_{(n-1)i}, x_{ni}), \widetilde{y}_i^*\right)$ $i = 1, \ldots, k$ with a minimum error [11]

$$\widetilde{y}_i = \widetilde{A}_n x_{ni} + \widetilde{A}_{n-1} x_{(n-1)i} + \cdots + \widetilde{A}_j x_{ji} + \cdots + \widetilde{A}_1 x_{1i} + \widetilde{A}_0. \tag{1}$$

In a more general situation, both inputs and outputs are treated as fuzzy numbers. In this case, we are looking for a function such as the one given by (2) to fit the fuzzy input-fuzzy output data $\left((\widetilde{x}_{1i}, \widetilde{x}_{2i}, \ldots, \widetilde{x}_{(n-1)i}, \widetilde{x}_{ni}), \widetilde{y}_i^*\right)$ [24]

$$\widetilde{y}_i = \widetilde{A}_n \widetilde{x}_{ni} + \widetilde{A}_{n-1} \widetilde{x}_{(n-1)i} + \cdots + \widetilde{A}_j \widetilde{x}_{ji} + \cdots + \widetilde{A}_1 \widetilde{x}_{1i} + \widetilde{A}_0 \tag{2}$$

where in (1) and (2), $\widetilde{A}_j (j = 0, \ldots, n)$ is the fuzzy coefficient (parameter) of the regression model. The parameters are optimized in such a way that the differences between the observed outputs $\widetilde{y}_i^*$ and estimated ones $\widetilde{y}_i$ are made as small as possible. All $\widetilde{y}_i$ $(i = 1, \ldots, k)$, $\widetilde{A}_j$ $(j = 0, \ldots, n)$, and $\widetilde{x}_{ji}$ are fuzzy numbers, and the operations used there are treated as the multiplication and addition of fuzzy numbers.

The first FLR model, proposed by Tanaka *et al.* [11], was based on the possibility theory and solved by linear programming (LP) [8], [11], [25]. Celmins [26] and Dimond [27] proposed an FR model based on the least-squares (LS) approach. According to the definition of the error, FR models are classified into two categories [8]: possibilistic with LP and LS approaches. In the former approach, the aim is to minimize the overall fuzziness by minimizing the total spread of the fuzzy coefficients, while the estimated outputs and the observed ones are within a certain h-level of confidence [7], [20]. The term h expresses the fitness between the estimated fuzzy outputs and the observed ones [7]. On the other hand, in the LS-based model, a similarity measure between observed and estimated outputs is used as the measurement for the fitness of the model [7], [8].

LP is a well-known technique used in possibilistic approaches to minimize the fuzziness of the coefficients. In the LP-based approaches with additional observation data, two constraints are added to LP [1]. However, having extra constraints is not an issue in practice since the datasets for FR are small, and LP problems even with thousands of variables and constraints can be solved in a few seconds.The Tanaka *et al.* LP approach [11] is criticized to be very sensitive to outliers [15], [20], [28], and if more input–output data are provided, then the spreads of estimated outputs may become undesirably wide [15], [20], [28]. However, considering the goal of LP-based approaches that is to cover the spreads of all the observations (up to an h-level), the sensitivity of these approaches to outliers or having wider spreads with more observations cannot be counted as drawbacks. Furthermore, in LP-based approaches, there is a tunable parameter by which the level of uncertainty might be kept under control. On the other hand, the goal of LS-based approaches that are the extension of the statistical regression analysis differs from LP-based approaches. The goal of LS-based approaches is to find a model that has the most similar estimated outputs to the observed ones based on the chosen similarity measure.

We propose an FLR model based on unconstrained global continuous optimization (UGCO). There are two main reasons for the application of UGCO to the design of the FR. First, we are interested in the ability to apply any type of model fitting. For example, a combination of covering all the observations (such as the ones for probabilistic regression) and minimizing the distance between the estimated and observed outputs (such as the LS-based approaches) can be minimized by multiobjective optimization methods. Second, we intend to seek a model with the minimum error of the chosen model-fitting measurement.

In this paper, we propose a metaheuristic UGCO method based on tabu search and harmony search for the FLR problem. The application of metaheuristic approaches in UGCO allows us not to be concerned about the differentiability of the given function—a condition that must be satisfied when dealing with analytical approaches. Therefore, any types of model fitting can be applied without being worried about how to model/solve problem or being concerned about differentiability or continuity of the chosen model fitting. The proposed method realizes the exploration of space by tabu search and then its further exploitation by harmony search. Tabu search is a robust technique to explore a wide area of the search space beyond the local optimality by positioning next solutions to unexplored areas [29]. It treats the objective function as a black box, which makes the optimization procedure context independent. Harmony search is also a context-independent search procedure that emulates some phenomena encountered in music. It employs a random search, which does not require the gradient of the function and hence makes it applicable to a wide range of optimization problems [30]. After finding a near-optimal solution, which is returned by tabu search, harmony search is then applied to exploit the surrounding area of such a near optimal solution to determine the global optimal solution.

The drawback of metaheuristic approaches is their slow rate of convergence [31]. In other words, they can find the near-minimal solution very quickly, but it is a time-consuming process to find a solution that is very close to the global minimum [31], [32], that is, the motivation to apply hybridized approaches that perform the diversification and then intensification in separate phases. In this paper, tabu search is applied for diversification, and improved harmony search is used for intensification. Tabu search is chosen for two reasons. First, it is a point-to-point approach that makes it less computationally expensive in terms of the number of function evaluations compared with population-based approaches such as genetic algorithms [32], [33]. Second, tabu search keeps track of the explored areas and stores them in a list so that it does not search the already explored areas. Thus, it is less likely for tabu search to trap into local minima or become stuck in specific areas for a long time [29]. In the intensification phase, an improved harmony search is applied. The advantage of harmony search over other metaheuristic approaches like genetic algorithms is that in harmony search the improvision is mostly made for each vector individually rather than mating two vectors to generate offsprings. The mating approach is suitable when we intend to find the near optimal solution, which in our approach is already sought in the diversification phase by tabu search.

To model and solve an FLR by the UGCO offers a number of benefits. When compared with the LS-based approaches (which are the extension of classical regressions) and the probabilistic approaches (which aim at covering all the data), the proposed approach can use any type of model fitting. For example, one may design an objective function that combines LS- and LP-based model-fitting measurements (MFMs). The application of UGCO does not necessarily increase the spread of estimated fuzzy outputs. However, the spreads may increase if a smaller error is obtained. Moreover, unlike in FR models such as those in [10], [24], [27], [34], and [35], the proposed method is not just restricted to triangular fuzzy numbers and can be used for any form of fuzzy numbers (different membership functions). The method is capable to find both positive and negative coefficients, while the methods that are presented in [3], [6], [24], [27], and [34] are unable to process negative coefficients.

The organization of this paper is as follows. As the main elements of the proposed UGCO method, a brief review of tabu search and harmony search is presented in Section II. This is followed by the proposal of UGCO method based on tabu and harmony search. In Section III, we discuss how the proposed UGCO method is used in the development of FR. In Section IV, we compare the results of the method to those existing in the literature over several well-known benchmark examples. Finally, we conclude the paper with a discussion of future work in Section V.

II. Global Continuous Optimization

The aim of UGCO methods is to search for the global minimum (maximum) of the given objective function. For a function of continuous variables X, we are interested in finding a vector of variables for which the function y attains its minimum. The optimality problem can be either concerned with finding the maximum or the minimum of the objective function. The mathematical formulation of UGCO for the minimization problem is given as follows:

$$y = \min f(X)$$
$$f : \mathbb{R}^n \rightarrow \mathbb{R}.$$

There are two different approaches in UGCO; analytical approaches and metaheuristic ones. In the former approaches, the objective function should be known and has to satisfy conditions such as differentiability, which is not necessary in metaheuristic-based approaches. In metaheuristic-based approaches, only the input–output behavior of the function has to be provided.

A. *UGCO Method*

Tabu search is a sound technique to find a near-global solution as it can help escape from the local minima and search in a given search space. However, it is reported that it is not efficient to search for a solution that is located close to the global minimum [31]. This fact motivates us to hybridize tabu search with other local optimizer methods to increase the effectiveness of the overall search process. Harmony search is applied in the second phase after finding the near-global optimal solution by tabu search. Lee and Geem [30] proposed harmony search as a new metaheuristic technique that uses the process of music generation. Similar to tabu search, harmony search is a stochastic random search method that does not require the gradient of the function, thus making it easier to be applicable in a wide range of optimization problems without being concerned with the constraints [30].

In this paper, we hybridize tabu search and the improved harmony search as a UGCO method. The improvement in harmony search aims at making it more suitable for local optimization. The near-optimum solution obtained by the tabu search is provided to the improved harmony search. In Sections II-A1 and A2, the details of tabu search (the first phase of the UGCO method) and the improved harmony search (the second phase of the UGCO method) are discussed.

1) First Phase (Tabu Search): Among the metaheuristic approaches, tabu search places a particular emphasis on escaping from traps of local minima. To accomplish this aim, tabu search keeps track of the last n visited solutions by placing them on a tabu list. Tabu search starts from a random solution positioned in the search space and then generates some neighbors around the current solution. Then, it checks the function's values for the neighbors. The next solution can even have worse performance compared with the current solution, and yet, it can be still selected. This allows tabu search to explore a wider search area. However, if the solution has been listed tabu, it can still be accepted as the next solution, provided that it is better than the best solution reported so far. The algorithm continues until it satisfies a certain stopping condition [29]. The stopping condition can be set to a maximum number of iterations. Alternatively, the algorithm can stop if the distance between the solutions in the kth and the $(k + n)$th iterations (where $k, n \epsilon \mathbb{N}$) is smaller than a small positive control value such as ϵ.

The starting solution always plays an important role in the rest of the optimization process. Starting from an initial point located far from the solution makes the optimization process very time consuming or even potentially trapped in a local minimum. To increase the speed of convergence and enhance the success rate, the initial solutions should be generated randomly to some extent but in a more systematic manner. Generat-

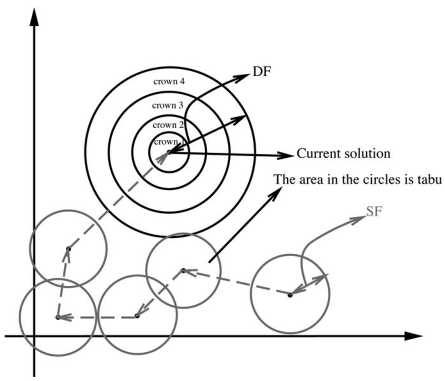

Fig. 1. Generating neighbors and tabulating areas based on DF and SF.

ing finite random solutions in a high-dimensional search space cannot normally cover the whole space. By virtue of that, we apply a partitioning process in which the search space is divided into partitions (sectors), and then, solutions are generated in each of the partitions in a random manner. This allows the solutions to be distributed all around the search space, while they are generated randomly in each partition. To assume that the domain of independent variables is $[a_i, b_i]$ (where i is the index of independent variables), we can divide each variable's domain into $m\epsilon\mathbb{N}$ equal intervals, and then, m^i random solutions are required to cover the partitions. The best solution is then selected as the current solution.

A move from the current solution to the next solution is carried out by a neighbor search strategy, which works based on the concept of randomization. In contrast with discrete tabu search, continuous tabu search has a more complicated neighbor search strategy as the number of neighbors around the current solution could be infinite. The basic mechanism is to generate random neighbors around the current solution with the maximum distribution radius of distribution factor (DF), and then, the best one is selected according to its fitness. To distribute the generated neighbors well, DF is divided into crowns, and one neighbor is generated for each of the crowns [31]. Thus, the neighbors are distributed homogeneously around the current solution. The crown neighbor search strategy enables the neighbors to deliver a better approximation of the entire neighborhood space.

In continuous tabu search, a solution is considered as tabu if it is in the neighborhood of a solution that has been seen before. The similarity factor (SF) defines the neighborhood radius of each solution. If SF is set to a small value, then tabu search works as a local optimizer. On the other hand, if SF is set to a large enough value, it behaves as a global optimizer. The process of generating random neighbors in the crowns with DF and tabulating the neighborhood space of unseen solution with SF is shown in Fig. 1.

In this paper, we consider two tabu lists: One for the short term and another one for the long term. A finite number of solutions recently visited are stored in the short-term tabu list; therefore, in the near future, we avoid searching them again. In each iteration, if a local-optimal solution is seen, we save it into the long-term tabu list. A solution is considered as a local-minimum solution if the next solution exhibits worse performance. Simply speaking, if the current solution is still improving, then we are getting close to the local optimum solution; otherwise, the current solution should be a local optimum one if the next one has worse performance. The long-term tabu list keeps track of local optimum solutions so that later, in the semi-intensification phase, the most potential one is searched.

Semi-intensification is carried on the stored solutions in the long-term tabu search to find out which of the local optima have the most potential to return a global optimum solution. The difference between semi-intensification and diversification is in applying smaller values for SF and DF. At the end of semi-intensification, only one point is nominated as the potential point for further intensification, which is followed by the improved harmony search.

2) Second Phase (Improved Harmony Search): Harmony search is a metaheuristic approach, which adopts the idea of natural musical processes [30]. The algorithm of harmony search minimizes an objective function of the form $f : \mathbb{R}^n \rightarrow \mathbb{R}$. In basic harmony search, randomly generated feasible solutions are initialized in the harmony memory (HM). In each iteration, the algorithm aims at improvising the HM. The improvision process works based on three operations: Memory consideration, pitch adjustment, and random selection. In the memory consideration and the random selection operations, each variable of a new solution vector is generated either based on the solutions in HM or randomly. The pitch adjustment operation, introduced to escape from local minima, makes random changes to some of the generated solutions [30], [36].

The algorithm of the basic harmony search can be outlined as follows [30]:

1) Initialization of control variables.
2) HM is initialized with harmony memory size (HMS) randomly generated feasible solution vectors X^i where $i = 1, 2, \ldots,$ HMS from the solution $X_t = (x_t^1, x_t^2, \ldots, x_t^{n-1}, x_t^n)$ obtained from the tabu search. The initial HM is made up of the solution sought from tabu search plus HMS $-$ 1 solutions that are chosen randomly in the neighborhood of X_t as follows:

for $i = 1$ to HMS $- 1$ **do**
  for $j = 1$ to n **do**
    $x_j^i \Leftarrow x_t^i + \mathrm{RND}(-0.5, 0.5)$;
  end for
end for

Then, the solutions are sorted according to the output that is produced by the objective function as follows:

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & \cdots & x_j^1 & \cdots & x_n^1 \\ x_1^2 & \cdots & x_j^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^i & \cdots & x_j^i & \cdots & x_n^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}-1} & \cdots & x_j^{\mathrm{HMS}-1} & \cdots & x_n^{\mathrm{HMS}-1} \\ x_1^{\mathrm{HMS}} & \cdots & x_j^{\mathrm{HMS}} & \cdots & x_n^{\mathrm{HMS}} \end{bmatrix}$$

where $X_j^i = (x_1^i, x_2^i, \ldots, x_j^i, \ldots, x_{(n-1)}^i, x_n^i)$ and n is the number of variables to be optimized.

3) A new HM is improvised from the current HM. So, for each solution vector X^i, $\acute{X}^i$ represents the new solution vector that is going to be substituted for X^i. The next solution can be either chosen from the HM with the probability of the harmony memory consideration rate (HMCR), or generated rondomly with the probability of $1-$ HMCR in the feasible search space M. In this paper, HMCR is set to be 0.9 as it is an intensification phase and we do not need too much mutation. This solution is then adjusted with a probability of pitch adjustment rate (PAR), and with the probability of $1-$ PAR, the solution remains intact. In pitch adjustment, the solution changes slightly in the neighborhood space of the solution. To be able to search the neighborhood of each solution comprehensively, we assign a large value like 0.8 to PAR. The entire process of improvising HM can be summarized as follows.

   HM consideration:

$$\acute{X}^i = \begin{cases} X^i \epsilon \text{HM}, & \text{with probability of HMCR} \\ X^i \epsilon M, & \text{with probability of (1-HMCR)} \end{cases}$$

   Pitch adjustment:

$$\acute{X}^i = \begin{cases} \acute{X}^i = \acute{X}^i + (bw \cdot \text{RND}(-1,1)) \\ \qquad \text{with probability of PAR} \\ X^i, \quad \text{with probability of } (1-\text{PAR}) \end{cases}$$

   where bw is an arbitrary positive value for pitch adjustment, which is usually assigned a value less than 1. In this paper, bw is set to 0.1. The function $\text{RND}(-1,1)$ generates a vector of random numbers coming from the uniform distribution over $[-1,1]$. The role of $bw \cdot \text{RND}(-1,1)$ is to produce both negative and positive values randomly. Therefore, if we assign an arbitrary value in the closed interval of $[-1,1]$ to bw instead of $[0,1]$, then we still need to produce random values in the interval of $[-1,1]$.

4) The new solution vector $\acute{X}^i$ is substituted for the worst solution in the HM–provided it outperforms the worst one.
5) If the stopping criterion is not met, then GOTO 3.

As the harmony search is applied in the intensification phase, we alter the improvision part of the basic harmony search and the one proposed in global-best harmony search [37]. First, to increase the processing speed, we do not consider the process of pitch adjustment for all solutions, which are made after the HM consideration phase. Thus, if a solution is selected randomly with the probability of $1-$ HMCR, then it does not need to undergo the pitch adjustment process [37]. Second, the pitch adjustment operation proposed by Omran and Mahdavi [37] based on the idea of particle swarm optimization is modified. In their approach, instead of making a random change in the generated solution after the HM consideration phase, the solution is replaced with the best solution in HM with the probability of PAR. We extend this approach by giving a chance not only to the best solution but to all the solutions. Here, in addition to extending the global best concept in [37], we have pitch adjustment as well. Therefore, in the proposed approach, if the pitch adjustment probability (PAP) is satisfied, then the solution is replaced with one of the solutions in HM with the probability

Algorithm 1 Proposed algorithm of the improvision step

for i = 1 to HMS **do**
  for j = 1 to n **do**
    if $RND(0,1) \leq HMCR$ **then**
      $\acute{X}_j^i \Leftarrow X_j^i$;
      if $RND(0,1) \leq PAR$ **then**
        if $RND(0,1) \leq BEST$ **then**
          Based on the Goodness Probability (GP) of solutions, select one of the solutions and store the index of this solution into b;
          $\acute{X}_j^i \Leftarrow X_b^i$;
        else
          $\acute{X}_j^i \Leftarrow \acute{X}_j^i + (bw \cdot RND(-1,1))$;
        end if
      end if
    else
      randomly select $\acute{X}_j^i \in M$;
    end if
  end for
end for

of an arbitrary control value *best*. The selection process of one of the solutions is based on the fitness of the solutions in the HM. According to their goodness, a probability of selection is assigned to each of the solutions. Thus, the goodness probability (GP) of a solution like X^i with a goodness value of $f(X^i)$ is computed as follows:

$$\text{GP}^i = \frac{f(X^i)}{\sum_{i=1}^{\text{HMS}} f(X^i)}.$$

Then, with the probability of $1-best$, the solution is randomly adjusted/tuned. As this is the intensification phase, we set $best = 0.8$ so that we have less random adjustment and more focus on local optimization. The improvision step of the proposed method is given in the form of Algorithm 1.

III. Fuzzy Linear Regression Based on Global Continuous Optimization

In this section, we show how to model the FLR as an optimization problem so that it can be solved by a UGCO. There are some studies in the field of FR by means of metaheuristic approaches. As an example, we can refer to genetic algorithms [13], [38], genetic programming [4], [39], tabu search [40], and fuzzy neural networks [41]–[43]. In this research, the FLR problem is addressed by the proposed UGCO method based on tabu search and improved harmony search. For the applications of tabu search in fuzzy optimization problems, see [29].

To find the optimized FLR model over the given numeric input–fuzzy output dataset $\left((x_{1i}, x_{2i}, \ldots, x_{(n-1)i}, x_{ni}), \widetilde{y}_i^*\right)$ [11] for $i = 1, \ldots, k$ with a minimum error can be viewed as a simple input–output system. This system receives the input vector and then produces the output based on the underlying function. The more general case of fuzzy input–fuzzy output [24] can be treated in the same way. Here, the underlying function should be the objective function (similarity measure) that shows the goodness of the FLR model. Thus, the input vector consists of the coefficients of the FLR model, and the output is the

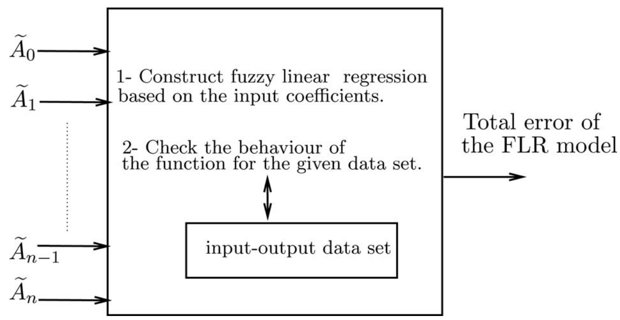

Fig. 2. Modeling of an FLR problem regarded as a UGCO problem.

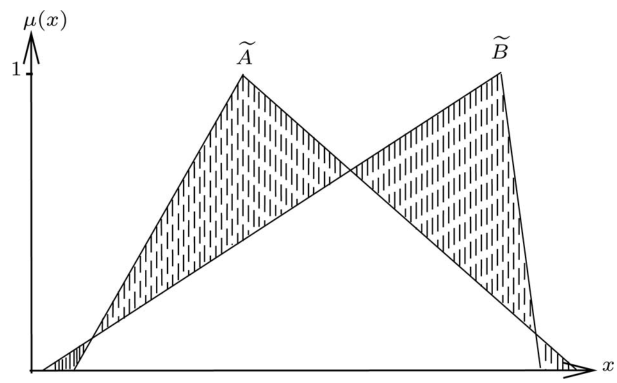

Fig. 3. Example of the NIA of two fuzzy numbers which is used in the numerator of the objective function given in (3).

generated error of this FLR model over the given data, as shown in Fig. 2.

To have the system presented in Fig. 2, the UGCO method starts with a random initial input vector that is basically the initial FLR model. After a number of iterations of guessing new input vectors, the UGCO method tends to find a near global optimum solution of the function. The input vector that enables the system to obtain its global minimum contains the optimized coefficients of the FLR model that fits the given data to the highest extent.

To show the fitness (performance) of the FLR model, we compare the fuzzy output of the model $\widetilde{y}_i$ with the observed one $\widetilde{y}_i^*$, where i is the index of the given data (point). To minimize the total difference between the observed fuzzy output and the estimated one is the goal of the optimization. There are different measures to determine the similarity between two fuzzy numbers. In this paper, we use one of the most used objective functions; the *relative nonintersected area* (RNIA) that computes the difference area of observed and estimated outputs, as given in (3) [8], [10], [28], [44], [45]. The numerator of the RNIA, the nonintersected area (NIA), for two fuzzy numbers is illustrated in Fig. 3. However, as mentioned earlier, the selection of RNIA is just for comparison purposes, and any other type of objective functions can be designed and then applied

$$\text{RNIA} = \sum_{i=1}^{k} \left[\frac{\int_{S_{\tilde{y}_i^*} \bigcup S_{\tilde{y}_i}} |\mu_{\tilde{y}_i(x)} - \mu_{\tilde{y}_i^*(x)}| dx}{\int_{S_{\tilde{y}_i^*}} \mu_{\tilde{y}_i^*(x)} dx} \right]. \quad (3)$$

Here, $S_{\tilde{y}_i^*}$ and $S_{\tilde{y}_i}$ are the supports of the observed output $\widetilde{y}_i^*$ and the estimated output $\widetilde{y}_i$, respectively. Note that if we consider the objective function in (3) alone, the possibility of being trapped in local minima is high. This happens as the method can be trapped at the zero-input vector (one of the potential local minima), where the error is equal to 1. The other drawback of RNIA is that if $\widetilde{y}^*$ and $\widetilde{y}$ do not exhibit any overlap, the estimated similarity becomes constant regardless of their relative position [28]. To avoid these mentioned drawbacks, we combine RNIA with the level-set difference measurement (LSM) given in [13], [21], and [44]. To describe the LSM procedure, we first start with preliminaries and pertinent notation.

We can break down the fuzzy coefficients into some level sets. This way, the input vector will have $(n+1) \times 2 \times \beta$ variables, where $n+1$ is the number of models coefficients, and $2 \times \beta$ is the number of left and right spreads of β level sets. For a fuzzy number like $\widetilde{A}$, where $\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) | x \epsilon \mathbb{R}, \mu_{\widetilde{A}} \rightarrow [0,1]\}$, the level set α (or α-cut) of $\widetilde{A}$ is as follows:

$$_{\alpha}A = \{x \epsilon \mathbb{R} | \ \mu_{\widetilde{A}}(x) \geq \alpha\}, \quad \alpha \epsilon (0,1]. \quad (4)$$

In (4), $_{\alpha}A$ can be represented as $[_{\alpha}a^L, _{\alpha}a^R]$, where $_{\alpha}a^L$ and $_{\alpha}a^R$ are the left and right spreads of $\widetilde{A}$ at level set α.

Now, we define the LSM measurement as follows:

$$\text{LSM} = \sum_{i=1}^{k} \left(\int_0^1 |_{\alpha}y_i^L - {}_{\alpha}y_i^{*L}| + |_{\alpha}y_i^R - {}_{\alpha}y_i^{*R}| \right) d\alpha. \quad (5)$$

Here, $_{\alpha}y_i^L, _{\alpha}y_i^{*L}$ and $_{\alpha}y_i^R, _{\alpha}y_i^{*R}$ are the left and right spreads of the estimated and observed fuzzy output at level set α, respectively. Note that in (5), we assume that $_0y_i^L$, $_0y_i^{*L}$, $_0y_i^R$, and $_0y_i^{*R}$ are equal to zero.

For a normalized fuzzy number with a triangular membership, function β is set to 2 as it can be represented by α-cuts of 0 and 1. We represent a triangular fuzzy number like $\widetilde{A}$ with the tuple (a^L, a^C, a^R), where a^C is the center of the membership function, and a^L, a^R are the left and the right spreads of the membership function. In this case, LSM that is given in (5) is simplified as follows:

$$\text{LSM} = \sum_{i=1}^{k} |y_i^L - y_i^{*L}| + |y_i^C - y_i^{*C}| + |y_i^R - y_i^{*R}|. \quad (6)$$

The extended objective function (EOF) that combines both RNIA and LSM to decrease the possibility to be trapped into local minima is given in the following form:

$$\begin{aligned} &\text{EOF}(\widetilde{A}_0, \widetilde{A}_1, \ldots, \widetilde{A}_n) \\ &= \sum_{i=1}^{k} \left[\int_0^1 \left(|_{\alpha}\widetilde{y}_i^L - {}_{\alpha}\widetilde{y}_i^{*L}| + |_{\alpha}\widetilde{y}_i^R - {}_{\alpha}\widetilde{y}_i^{*R}| \right) d\alpha \right] \\ &+ \sum_{i=1}^{k} \left[\int_{S_{\tilde{y}_i^*} \bigcup S_{\tilde{y}_i}} |\mu_{\tilde{y}_i(x)} - \mu_{\tilde{y}_i^*(x)}| dx \right]. \quad (7) \end{aligned}$$

The MFM given in (7) works only for numeric input–fuzzy output or fuzzy input–fuzzy output. However, other MFMs can be applied to derive a linear regression model for numeric inputs and outputs.

To come up with a sound approximation for the search space (domain) of each coefficient, we apply statistical linear regression as a preprocessing phase. The given data are defuzzified (decoded to numeric values), and then, the best linear function is sought to fit them. In defuzzification, we simply take the center point of the core for each fuzzy number. The neighborhood of the coefficients of the linear regression are later taken as the search domain for each fuzzy coefficient in FLR.

The objective function to find the optimized triangular normalized fuzzy numbers as the coefficients of the FLR model is defined as follows:

$$\begin{aligned} y &= \min \text{EOF}(\widetilde{A}_0, \widetilde{A}_1, \ldots, \widetilde{A}_n) \\ &= \min \text{EOF}(a_0^R, a_0^C, a_0^L, a_1^R, a_1^C, a_1^L \cdots, a_n^R, a_n^C, a_n^L) \\ \text{EOF} &: [(k_1-\theta)-\gamma, (k_1-\theta)] \times [k_1-\theta, k_1+\theta] \\ &\quad \times [k_1+\theta, (k_1+\theta)+\gamma] \times \cdots \times [(k_n-\theta)-\gamma, (k_n-\theta)] \\ &\quad \times [k_n-\theta, k_n+\theta] \times [k_n+\theta, (k_n+\theta)+\gamma] \to \mathbb{R} \end{aligned} \tag{8}$$

where $k_0, k_1, \ldots, k_n$ are the coefficients of the linear function that fits the defuzzified data, and θ and γ are neighborhood constants that define the domain of the fuzzy coefficients of FLR.

IV. Experiments

In this section, we apply the UGCO method to the development of the FLR model. The datasets that are taken from the literature are divided into two main classes: numeric input–fuzzy output data and fuzzy input–fuzzy output data. These two classes just involve symmetric fuzzy numbers.

Since the proposed method is not just restricted to symmetric fuzzy numbers, we have validated our method in terms of asymmetric fuzzy input–asymmetric fuzzy output as the third class. Moreover, the performance of the model is investigated for a large dataset with outliers.

Different models in the literature applied different MFMs such as the Hojati *et al.* similarity measure [34], distance criterion [46], Euclidean distance [47], NIA [3], [6], relative NIA [10], and compatibility measure [48]. Since there is no evidence that one measure is better than the others, one may adopt an MFM based upon requirements of modeling and the nature of the dataset. Thus, in fairness to other studies in the literature, three other reported MFMs are applied for comparative analysis of models' performance. The first one is a similarity measure proposed by Hojati *et al.* [34] given by (9), the second one is a distance criterion that is proposed in [46] expressed as (10), and the third one is the NIA proposed in [3] and [6] and defined by (11)

$$S_h = \frac{\int \min(\mu_{\tilde{y}^*(x)}, \mu_{\tilde{y}(x)})dx}{\int \max(\mu_{\tilde{y}^*(x)}, \mu_{\tilde{y}(x)})dx} \tag{9}$$

$$D = \frac{1}{4}\sum_{i=1}^{k} |y_i^L - y_i^{*L}| + 2|y_i^C - y_i^{*C}| + |y_i^R - y_i^{*R}| \tag{10}$$

$$\text{NIA} = \sum_{i=1}^{k}\left[\int_{S_{\tilde{y}_i^*} \bigcup S_{\tilde{y}_i}} |\mu_{\tilde{y}_i(x)} - \mu_{\tilde{y}_i^*(x)}|dx\right]. \tag{11}$$

Note that Hojati's original MFM (9) is a similarity measure with values between 0 and 1; therefore, the closer the value to 1, the better the model. However, for the sake of conformity with other MFMs, we have reported $1 - S_h$ that can be seen as a dissimilarity measure, and the closer the value to 0, the better the model.

For comparison of the models for a given dataset, we say model f_1 is superior to model f_2 over $\text{MFM}_1, \text{MFM}_2, \ldots, \text{MFM}_n$, where all MFMs are minimization problems, only if all the following *"FR model comparison conditions"* simultaneously hold:

$$\begin{gathered} \text{MFM}_1(f_1) \le \text{MFM}_1(f_2) \\ \text{MFM}_2(f_1) \le \text{MFM}_2(f_2) \\ \vdots \\ \text{MFM}_n(f_1) \le \text{MFM}_n(f_2). \end{gathered} \tag{12}$$

In the case of a dataset where only some of the aforementioned conditions hold, it can be said that f_1 and f_2 are *"Pareto-equivalent."* Note that in some cases, f_1 can be superior to f_2 with respect to $\text{MFM}_1, \text{MFM}_2, \ldots, \text{MFM}_n$, however, if, for a new MFM', f_2 is superior to f_1, then f_1 and f_2 are said to be Pareto-equivalent. Therefore, to compare the models for a dataset, one should consider the client's requirements and requests and the nature of the dataset to select suitable MFMs. However in this study, we restricted our comparisons based on RNIA, NIA, distance criterion, and Hojati's measure as the applied datasets are mainly used for benchmarking purposes.

Four criteria given by (3) and (11), distance criterion and Hojati's measure are reported for all the experiments. The similarity measure (11) is the total NIA between the observed and the estimated fuzzy outputs. Note that we have used (7) as the main criterion to carry out the optimization process; however, the FR model comparison conditions are applied to compare the models in terms of all the four measures. From now on, in the results, the values with and without rounded parentheses $(*)$ show the corresponding values of NIA and RNIA, respectively.

All metaheuristic approaches require that one should set their control variables. The setting of the parameters is usually done by trial and error or based on some experts' intuitions. As an example, in genetic algorithms, one should set the mutation and crossover percentages, elite count, population size, selection method, etc. For the proposed method the parameters' settings for all the experiments are given in Table I. To start, k_i in (8) that are the coefficients of the linear regression to fit the defuzzified given data are computed. For defuzzification, the mean of the inputs and outputs' core are considered. Then, we generate initial population size (IPS) random fuzzy solutions around the coefficients. Intuitively, the fuzzy coefficients for FLR should be close to k_i; therefore, we set our search space (domain) for FLR in (8) with small vales like 4 and 2 for "cores neighborhood constant" θ and "spread neighborhood constant" γ, respectively.

TABLE I
PARAMETER SETTINGS FOR BOTH PHASES OF THE PROPOSED METHOD

Diversification phase with TS	
Parameter	***Value***
Cores neighborhood space (θ)	4
Spreads neighborhood space (γ)	2
Initial population size (IPS)	64
Population size	5
Number of iterations with no improvement (NI-first)	5
Tabu list size	10

Intensification phase with Improved HS	
Parameter	***Value***
Population size	10
Neighborhood space (NS-second)	±0.5
Depth (Dep)	8
Number of iterations with no improvement (NI-second)	5
Decrement coefficient of neighborhood space (Dec)	2

The IPS plays an important role to arrive at a sound final result. By performing preliminary experiments, we noticed that any population size more than 50 produces good results; therefore, we set it to be between 60 to 70. A very large value for population size does not increase the performance considerably as much as they slow down the optimization process. The diversification phase runs until it finds a better result for five consecutive iterations. We have observed that if we set *NI-first* to more than 5, the result will not improve significantly, but it just makes the diversification phase very time consuming. Similarly, to set *NI-first* less than 5 increases the possibility that it will be trapped in local minima.

In the intensification phase, we define a small search space as we already found the near optimum solution; therefore, *NS-second* is set to 0.5. With the trial-and-error method, we found if the value of *NS-second* is greater than 1, then the intensification phase does more diversification, and the use of values smaller than 0.1 makes the intensification very time consuming. In contrast with the diversification phase, in the intensification phase, the neighborhood space gets smaller if there is no improvement after *NI-second*. We set "decrement coefficient of neighborhood space" (*Dec*) to 2 as in the final stages, the search space should get very small so the algorithm can find a solution very close to the optimum. The search space can become smaller up to "depth" times, and therefore, in the final stage of intensification, the search space is as small as $\frac{NS-second}{Dec^{Dep}} = \frac{\pm 0.5}{2^8}$, which, in here, is equal to ± 0.0019. As mentioned earlier, the majority of the values shown in Table I are selected in an intuitive fashion or/and by the trial-and-error method.

As the proposed hybrid optimization method is nondeterministic, 1000 separate runs with random seeds are attempted for each dataset. The average, best, and worst cases are then reported. In the literature of global continuous optimization with metaheuristics, the speeds of the algorithms are compared based on their number of function evaluations rather than the execution time. This is due to the fact that the execution time of a method strictly depends on the computer's speed, the code optimization, and the programming platform [32], [49]. Therefore, for fair comparison between the speeds of different global continuous optimization methods, only the number of function evaluations is considered. However, in this paper, to show an approximation of how long the optimization process takes to compute a model, the execution time is given just for a couple of the datasets. The simulations were run on a PentiumIV 3.00-GHz computer with 2 GB of memory. The program is coded in MATLAB7.0.

TABLE II
TANAKA *ET AL.* DATASET [52]

Obs.	Independent variable	Dependent variable
1	1	(6.2, 8.0, 9.8)
2	2	(4.2, 6.4, 8.6)
3	3	(6.9, 9.5, 12.1)
4	4	(10.9, 13.5, 16.1)
5	5	(10.6, 13.0, 15.4)

TABLE III
EXPERIMENTAL RESULT FOR 1000 RUNS FOR THE DATASET GIVEN IN TABLE II BY THE PROPOSED METHOD

	best solution	worst solution	average	STD
Error (7)	11.7473	12.5346	12.0154	0.1760
Error (11)	8.3678	9.0414	8.5736	0.1426
Iterations	231	213	224	26
CPU time (s)	3.3326	3.0395	3.0811	0.2814

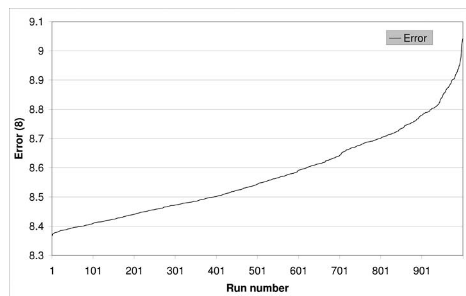

Fig. 4. Sorted generated errors (11) for 1000 separate runs for dataset given in Table II obtained by the proposed method.

A. Numeric Input–Symmetric Fuzzy Output Results

Example 1: Tanaka *et al.* in the first work on FLR defined the dataset given in Table II to investigate the FLR model's performance. This dataset has been used in many research studies as a benchmark, see [3], [6], [8], [10]–[12], [27], [28], [34], [45], and [50]. This dataset is an example of a small size numeric input–fuzzy output data. The results for 1000 separate runs with random initial solutions are reported in Table III. The generated errors produced by the estimated models for 1000 runs are sorted and reported in Fig. 4. The FR model comparison conditions and the results given in (12) and Table III reveal that our model is Pareto-equivalent to the models given by Hojati *et al.* [34], Mashinchi *et al.* [40], Lu and Wang [20], and Hassanpour [51] and superior to the rest of the models over all of RNIA, NIA, distance criterion, and Hojati's measure.

The models produced by the method presented here and other methods for the dataset shown in Table II are given in Table IV. The proposed method has the smallest total error (11) and outperforms the other methods reported in the literature based on the error given in (11). The plot of the estimated fuzzy function of some models for the given data in Table II is illustrated in Fig. 5.

TABLE IV
COMPARATIVE ANALYSIS OF FUZZY MODELS IN TABLE II

Method	A_0	A_1
This work (best solution)	$(5.0798, 6.7652, 7.0613)$	$(1.1042, 1.2472, 1.6679)$
Chen and Hsueh (2009) [28]	$(2.63, 4.95, 7.27)$	$(1.71, 1.71, 1.71)$
Mashinchi et al. (2009) [40]	$(5.07, 6.72, 8.38)$	$(1.12, 1.27, 1.38)$
Lu and Wang (2009) [20][1]	$(4.25 + 1.75x_i, -0.10x_i + 2.90)$	
Shakouri and Nadimi (2009) [8]	$(3.2, 5.042, 6.884)$	$(1.480, 1.592, 1.704)$
Hassanpour (2008) [51]	$(4.95, 6.75, 8.55)$	$(1.05, 1.25, 1.45)$
Bargiela and Pedrycz (2007) [12]	$(4.95, 4.95, 4.95)$	$(1.719, 1.719, 1.719)$
Hojati, Bector and Smimou (2005) [34]	$(5.1, 6.75, 8.4)$	$(1.10, 1.25, 1.40)$
Modaress et al. (2005) [45]	$(2.98, 4.82, 6.66)$	$(1.50, 1.66, 1.82)$
Nasrabadi and Nasrabadi (2004) [35]	$(2.36, 4.68, 7.00)$	$(1.73, 1.73, 1.73)$
Kao and Chyu (2003) [6]	$(2.606, 4.926, 7.246)$	$(1.718, 1.718, 1.718)$
Kao and Chyu (2002) [3]	$(1.94, 4.95, 6.75)$	$(1.71, 1.71, 1.71)$
Özelkan and Duckstein (2000) [53][2] ($v = 25$)	$(3.4, 5.9, 8.4)$	$(1.4, 1.4, 1.4)$
Kim and Bishu (1998) [10]	$(3.11, 4.95, 6.84)$	$(1.55, 1.71, 1.82)$
Savic and Pedrycz (1991) [50][2]	$(0.92, 4.95, 8.98)$	$(1.64, 1.71, 1.78)$
Tanaka et al. (1989) [52][2]	$(0, 3.85, 7.7)$	$(2.1, 2.1, 2.1)$
Diamond (1988) [27][2]	$(3.11, 4.95, 6.79)$	$(1.55, 1.71, 1.87)$
Tanaka et al. (1982) [11]	$(0, 3.84, 7.69)$	$(2.1, 2.1, 2.1)$

[1]The estimated model for [20] has variable spread which we have shown it by $\widetilde{E_i}$.
[2]The estimated model for [27], [50], [52] and [53] are taken from [3], [10], [10], and [34], respectively.

TABLE V
AVERAGE AND GENERATED ERROR OF THE ESTIMATED MODELS FOR THE GIVEN DATASET IN TABLE II

Obs.	This work	Hojati et al. (2005) [34]	Mashinchi et al. (2009) [40]	Lu and Wang (2009) [20]	Shakouri and Nadimi (2009) [8]	Hassanpour (2008) [51]
1	0.5365 (0.2981)	0.0000 (0.0000)	0.03 (0.02)	3.13 (1.74)	2.2349 (1.24)	0.200 (0.1111)
2	3.3419 (1.5190)	3.7428 (1.7013)	3.71 (1.69)	2.33 (1.06)	2.8704 (1.30)	3.854 (1.7518)
3	1.5218 (0.5853)	1.7872 (0.6874)	1.78 (0.69)	0 (0)	0.6405 (0.27)	1.800 (0.6923)
4	2.9654 (1.1405)	2.8686 (1.1033)	2.86 (1.10)	3.51 (1.35)	3.2867 (1.26)	2.911 (1.1196)
5	0.0022 (9.16e-4)	0 (0)	0.12 (0.05)	0 (0)	0.0040 (0.00)	0.400 (0.1667)
Ave_e	1.6736 (0.7088)	1.6797 (0.6984)	1.70 (0.71)	1.79 (0.83)	1.8073 (0.8160)	1.8330 (0.7683)
NIA	8.3678	8.3986	8.51	8.96	9.0365	9.165
RNIA	3.5438	3.492	3.54	4.15	4.08	3.8415
Tot_e (Hojati's criterion)	0.494	0.4485	0.4642	0.4538	0.5186	0.4848
Tot_e (distance criterion)	5.6310	5.6	5.6825	5.6	5.6540	5.9

Obj.	Özelkan and Duckstein (2000) [53]	Modarres et al. (2005) [45]	Kao and Chyu (2002) [3]	Nasrabadi and Nasrabadi (2004) [35]	Chen and Hsueh (2009) [28]
1	1.286 (0.714)	2.4320 (1.35)	2.7862 (1.5479)	2.564 (1.426)	2.324 (1.290)
2	3.474 (1.579)	2.7856 (1.27)	2.5901 (1.1773)	2.813 (1.277)	3.081 (1.400)
3	1.130 (0.434)	0.5817 (0.23)	0.5555 (0.2137)	0.718 (0.273)	1.092 (0.418)
4	3.216 (1.237)	3.2608 (1.25)	3.3604 (1.2925)	3.062 (1.179)	2.526 (0.976)
5	0.198 (0.082)	0.2971 (0.13)	0.3853 (0.1605)	0.614 (0.265)	0.947 (0.392)
Ave_e	1.860 (0.809)	1.8714 (0.8460)	1.9355 (0.8784)	1.954 (0.884)	1.996 (0.896)
NIA	9.303	9.3572	9.6776	9.771	9.976
RNIA	4.047	4.23	4.3919	4.421	4.477
Tot_e (Hojati's criterion)	0.5048	0.5330	0.5337	0.5586	0.5886
Tot_e (distance criterion)	5.7	5.78	6.1025	5.933	6.1

Obj.	Kim and Bishu (1998) [10]	Kao and Chyu (2003) [6]	Diamond (1988) [27]	Tanaka et al. (1989) [52]	Tanaka et al. (1982) [11]	Savic and Pedrycz (1991) [50]
1	2.207 (1.226)	2.217 (1.259)	2.2075 (1.2264)	3.364 (1.869)	3.3562 (1.8646)	2.7764 (1.54)
2	3.025 (1.375)	3.024 (1.396)	3.0499 (1.3863)	2.840 (1.291)	2.8500 (1.2955)	3.3308 (1.52)
3	1.041 (0.401)	1.082 (0.420)	1.0916 (0.4198)	1.512 (0.581)	1.5225 (0.5856)	1.7959 (0.7)
4	2.902 (1.116)	2.812 (1.083)	2.8444 (1.0940)	2.269 (0.873)	2.2578 (0.8684)	2.9968 (1.16)
5	0.850 (0.354)	0.954 (0.406)	0.9504 (0.3960)	2.404 (1.002)	2.4153 (1.0064)	2.0694 (0.86)
Ave_e	2.004 (0.894)	2.178 (0.913)	2.0288 (0.9045)	2.478 (1.123)	2.4803 (1.1241)	2.5939 (1.1560)
NIA	10.026	10.089	10.1438	12.388	12.4017	12.9693
RNIA	4.472	4.564	4.5225	5.616	5.6203	5.78
Tot_e (Hojati's criterion)	0.5874	0.5905	0.5913	0.5687	0.5685	0.5631
Tot_e (distance criterion)	6.05	6.11	6.1	7.3	7.29	7.85

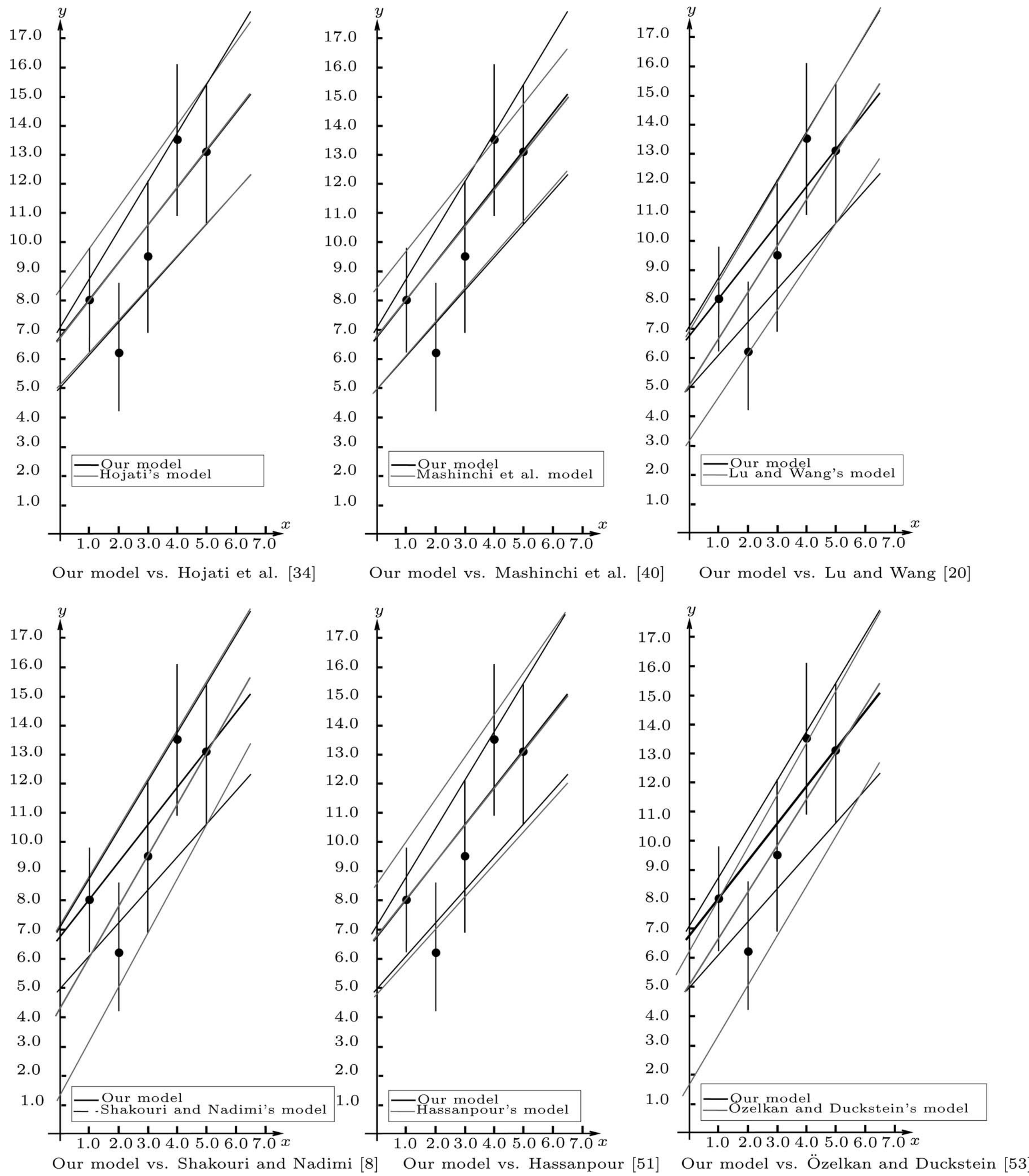

Fig. 5. Estimated fuzzy function for the data given in Table II. A comparative analysis involves six other models available in the literature.

TABLE VI
KIM AND BISHU'S DATASET [10]

Obs.	Dependent variable *(Response time)*	Independent variable *(Inside control room experience)*	Independent variable *(Outside control room experience)*	Independent variable *Education*
Team 1	$(5.83, 3.56)$	2.0	0.0	15.25
Team 2	$(0.85, 0.52)$	0.0	5.0	14.13
Team 3	$(13.93, 8.5)$	1.13	1.5	14.13
Team 4	$(4, 2.44)$	2.0	1.25	13.63
Team 5	$(1.65, 1.01)$	2.19	3.75	14.75
Team 6	$(1.58, 0.96)$	0.25	3.5	13.75
Team 7	$(8.18, 4.99)$	0.75	5.25	15.25
Team 8	$(1.85, 1.13)$	4.25	2.0	13.5

TABLE VII
EXPERIMENTAL RESULT FOR 1000 RUNS FOR THE DATASET GIVEN IN TABLE VI

	best solution	worst solution	average	STD
Error (7)	29.7202	34.6411	12.0154	1.2071
Error (11)	21.9369	25.8395	23.7067	0.4984
Iterations	300	287	301	24
CPU time (s)	5.1921	5.1049	5.0826	0.327

Example 2: The dataset in this example given in Table VI is related to cognitive response times of the nuclear power plant control room crew to an abnormal event and has been introduced

TABLE VIII
MODELS AVAILABLE IN THE LITERATURE FOR THE DATASET IN TABLE VI

Method	Estimated function
This work	$(-16.44, -16.25, -16.09) + (-0.77, -0.31, -0.12)x_1 + (-1.17, -1.14, -1.13)x_2 + (1.60, 1.61, 1.62)x_3$
Shakouri and Nadimi (2009) [8]	$(-20.08, -20.08, -20.08) + (-0.23, -0.16, -0.09)x_1 + (-1.22, -0.9, -0.58)x_2 + (1.66, 1.81, 1.96)x_3$
Modarres et al. (2005) [45][1]	$(-23.0624 - 10.0507x_1 - 1.6288x_2 + 2.3474x_3, -10.2731 - 0.6696x_1 - 0.7199x_2 + 1.1341x_3)_L$
Kim and Bishu (1998) [10]	not reported
Savic and Pedrycz (1991) [50]	not reported
Tanaka et al. (1989) [52]	not reported

[1]This model is represented in the form of $(y_i^c, \alpha_{y_i})_L$ which is equal to $(y_i^c - \alpha_{y_i}, y_i^c, y_i^c + \alpha_{y_i})$.

TABLE IX
AVERAGE AND GENERATED ERROR OF THE ESTIMATE MODELS FOR THE GIVEN DATASET IN TABLE VI

	This work	Shakouri and Nadimi[4] (2009) [8]	Kim and Bishu[3] (1998) [10]	Modarres et al. (2005) [45]	Tanaka et al.[3] (1989) [52]	Savic and Pedrycz[3] (1991) [50]
Team 1	3.07 (0.86)	2.43 (0.68)	5.64 (1.58)	7.52 (2.11)	9.80 (2.75)	10.29 (2.89)
Team 2	0.10 (0.19)	3.20 (6.16)	2.66 (5.13)	2.20 (4.23)	8.87 (17.07)	9.86 (18.97)
Team 3	9.23 (1.09)	11.05 (1.30)	10.48 (1.23)	10.09 (1.18)	2.78 (0.32)	7.89 (0.92)
Team 4	1.49 (0.61)	1.56 (0.64)	1.53 (0.63)	2.10 (0.86)	10.46 (4.29)	8.19 (3.35)
Team 5	1.15 (1.14)	2.82 (2.79)	2.84 (2.81)	2.69 (2.66)	6.49 (6.42)	9.56 (9.47)
Team 6	0.56 (0.58)	2.24 (2.33)	3.37 (3.51)	2.92 (3.04)	10.57 (11.01)	10.02(10.44)
Team 7	5.69 (1.14)	7.65 (1.53)	5.62 (1.12)	6.30 (1.26)	4.90 (0.98)	5.83 (1.16)
Team 8	0.63 (0.56)	1.83 (1.62)	1.05 (0.93)	1.02 (0.90)	6.73 (5.95)	7.49 (6.63)
NIA	21.93	32.78[2]	33.25	34.85	60.64[1]	69.17[1]
Tot_e RNIA	6.18	17.06[2]	16.98	16.17	48.82[1]	53.87[1]
T_e (Hojati's criterion)	0.6521	0.7382	0.8217	0.84	0.7989	0.8017
T_e (Distance criterion)	20.3129	23.3340	23.6700	24.0550	42.9700	41.9500

[1]The estimated models error of Savic91 [50] and Tanaka89 [52] were reported in [10].
[2]According to the estimated function given in Table VIII which is taken from [8], Shakouri's model error will be NIA = 32.78 and RNIA = 17.06. However, the error was reported RNIA = 5.16 in Shakouri09 [8]–which is incorrect.
[3]The estimated outputs are taken from Modaress *et al.* [45] as the estimated models are not reported in the original papers.
[4]The reported fuzzy outputs and their corresponding errors from Shakouri [8] and the corrected ones are given in Table XXI in the Appendix.

by Kim and Bishu [10]. This is a benchmark dataset for fuzzy multiple linear regression. The crew's fuzzy response time (in minutes) to an abnormal event is dependent on crews' experience inside a control room, experience outside a control room, and training. This dataset has been applied as a benchmark for comparison of the methods presented in [8], [10], and [45]. The estimated model of this study in comparison with other works is given in Table VIII. Based on (12) and the results of comparative analysis shown in Table IX, our method is superior to the other models over RNIA, NIA, distance criterion, and Hojati's measure.

The results given in Table VII shows the error and the number of iterations for the case of the best and the worst solutions for 1000 separate runs. As an example, Fig. 6 illustrates the convergence of the error for the best run. It can be observed that the first phase is carried out for 125 iterations, and then, the intensification phase is started. In the first phase, the average error in each iteration exhibits some oscillations, which is due to the nature of tabu search applied in the diversification phase. The oscillations show the effectiveness of tabu search that pushes the next solutions to an area that has not been searched so far and does not necessarily exhibit higher potential. By doing this, we can be almost sure that during the diversification phase, we have effectively searched the search space. In the intensification phase, the average error in each iteration decreases smoothly with no oscillations as in each iteration, we get closer to the actual optimal solution.

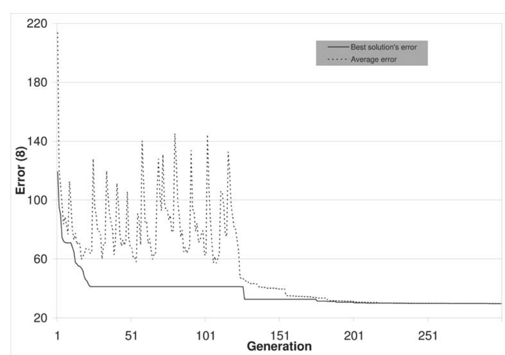

Fig. 6. Convergence of the error for the best model over the dataset in Table VI.

Example 3: The dataset in Table X was proposed by Tanaka *et al.* [11] and is concerned with the price of housing in Japan. This gives rise to the fuzzy multiple regression analysis as the price variable is dependent on five numeric variables: the rank of material (x_1), the first floor space in square meters (x_2), the second floor space in square meters (x_3), the number of rooms (x_4), and the number of Japanese-style rooms (x_5), given in Table X. This dataset has the largest number of records in comparison with other datasets and is a good benchmark for fuzzy multiple linear regression and negative coefficients.

This dataset has been applied to study the efficiency of the FLR methods presented in [11], [17] and [54]–[56]. Table X shows the error produced by each method in comparison with

TABLE X
TANAKA *ET AL.* HOUSE PRICE DATASET [11]

Obs.	Dependent variable $\widetilde{y}^*$	Independent variable $(x_1, x_2, x_3, x_4, x_5)$	Model's error This work	Kao and Chyu (2005) [55]	Tanaka et al. (1982) [11]
1	$(6060, 550)$	$(1, 38.09, 36.43, 5, 1)$	390.27 (0.7096)	577.5 (1.05)	187 (0.34)
2	$(7100, 50)$	$(1, 62.10, 26.50, 6, 1)$	204.71 (4.0942)	431.5 (8.63)	663.5 (13.27)
3	$(8080, 400)$	$(1, 63.76, 44.71, 7, 1)$	252.56 (0.6314)	132 (0.33)	528 (1.32)
4	$(8260, 150)$	$(1, 74.52, 38.09, 8, 1)$	182.24 (1.2150)	307.5 (2.05)	591 (3.94)
5	$(8650, 750)$	$(1, 75.38, 41.40, 7, 2)$	541.30 (0.7217)	405 (0.54)	7.5 (0.01)
6	$(8520, 450)$	$(2, 52.99, 26.49, 4, 2)$	609.20 (1.3538)	558 (1.24)	625.5 (1.39)
7	$(9170, 700)$	$(2, 62.93, 26.49, 5, 2)$	664.66 (0.9495)	455 (0.65)	259 (0.37)
8	$(10310, 200)$	$(2, 72.04, 33.12, 6, 3)$	2.78 (0.0139)	252 (1.26)	912 (4.65)
9	$(10920, 600)$	$(2, 76.12, 43.06, 7, 2)$	606.42 (1.0107)	558 (0.93)	486 (0.81)
10	$(12030, 100)$	$(2, 90.26, 42.64, 7, 2)$	328.48 (3.2848)	500 (5.00)	947 (9.47)
11	$(13940, 350)$	$(3, 85.70, 31.33, 6, 3)$	328.81 (0.9395)	276.5 (0.79)	990.5 (2.83)
12	$(14200, 250)$	$(3, 95.27, 27.64, 6, 3)$	52.94 (0.2118)	295 (1.18)	820 (3.28)
13	$(16010, 300)$	$(3, 105.98, 27.64, 6, 3)$	505.30 (1.6844)	696 (2.32)	1104 (3.68)
14	$(16320, 500)$	$(3, 79.25, 66.81, 6, 3)$	251.65 (0.5033)	486 (0.81)	530 (1.06)
15	$(16990, 650)$	$(3, 120.5, 32, 25, 6, 3)$	524.19 (0.8065)	273 (0.42)	780 (1.20)
Average			361.06 (1.23)	413.53 (1.81)	616.26 (3.17)
NIA			5445.6	6203	9244
RNIA			18.13	27.20	47.62
T_e (Hojati's criterion)			0.7055	0.6763	0.7294
T_e (Distance criterion)			4.5835e+003	4.2524e+003	7.1557e+003

TABLE XI
EXPERIMENTAL RESULT FOR 1000 RUNS FOR THE DATASET GIVEN IN TABLE X

	best solution	worst solution	average	STD
Error (7)	6355.18	8645.13	7098.08	354.67
Error (11)	5415.95	7539.95	6140.47	314.46
Iteration	406	410	426	25

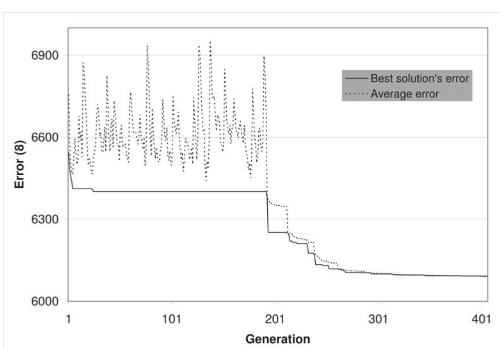

Fig. 7. Convergence of the error for the best model over the dataset in Table X.

the model that is given in (13). According to the error analysis reported in Table X, the model presented here is Pareto-equivalent to the Kao and Lin's model [55] and superior to the Tanaka *et al.* model [11]

$$
\begin{aligned}
y = {} & (2429.0, 2430.6, 2458.4)x_1 + (83.46, 83.48, 83.55)x_2 \\
& + (82.27, 84.69, 87.21)x_3 + (-402.54, -386.891, \\
& - 383.892)x_4 + (-222.55, -220.64, -206.89)x_5 \\
& + (-1134.7, -1072.0, -1042.9). \qquad (13)
\end{aligned}
$$

The performance of the method reported for 1000 separate runs is given in Table XI. The convergence error for the best model is given in Fig. 7. In the first 200 generations, the method is concerned with the diversification phase, and here, we observe some oscillations because of searching in different parts of the search domain.

TABLE XII
HONG AND HWANG'S DATASET [14]

Obs.	Independent variable	Dependent variable
1	1	$(-1.6, 0.5)$
2	3	$(-1.8, 0.5)$
3	4	$(-1.0, 0.5)$
4	5.6	$(1.2, 0.5)$
5	7.8	$(2.2, 1.0)$
6	10.2	$(6.8, 1.0)$
7	11.0	$(10.0, 1.0)$
8	11.5	$(10.0, 1.0)$
9	12.7	$(10.0, 1.0)$

TABLE XIII
EXPERIMENTAL RESULT FOR 1000 RUNS FOR THE DATASET GIVEN IN TABLE XII BY THE PROPOSED METHOD

	best solution	worst solution	average	STD
Error (7)	9.96	10.51	10.27	0.37
Error (11)	6.89	7.30	7.03	0.07
Iteration	229	244	210	24

TABLE XIV
SAKAWA AND YANO'S DATASET [24]

Obs.	Independent variable	Dependent variable
1	$(1.5, 2, 2.5)$	$(3.5, 4.0, 4.5)$
2	$(3.0, 3.5, 4.0)$	$(5.0, 5.5, 6.0)$
3	$(4.5, 5.5, 6.5)$	$(6.5, 7.5, 8.5)$
4	$(6.5, 7.0, 7.5)$	$(6.0, 6.5, 7.0)$
5	$(8.0, 8.5, 9.0)$	$(8.0, 8.5, 9.0)$
6	$(9.5, 10.5, 11.5)$	$(7.0, 8.0, 9.0)$
7	$(10.5, 11.0, 11.5)$	$(10.0, 10.5, 11.0)$
8	$(12.0, 12.5, 13.0)$	$(9.0, 9.5, 10.0)$

Example 4: The dataset given in Table XII is proposed by Hong *et al.* [14]. What makes this dataset different from the others is that negative fuzzy outputs are also considered. The proposed model and the comparison of the resulting error with

TABLE XV
ESTIMATED MODELS AND GENERATED ERRORS FOR SEVERAL MODELS AND DATA SHOWN IN TABLE XIV

Method	A_0	A_1	NIA	RNIA	T_e (Hojati's criterion)	T_e (Distance) criterion)
This work	$(2.9606, 2.9679, 3.0042)$	$(0.5154, 0.5154, 0.5154)$	5.265	8.3046	0.7508	6.2460
Mashinchi et al. (2009) [40]	$(3.00, 3.02, 3.20)$	$(0.50, 0.50, 0.50)$	5.330	8.4608	0.7093	6.2050
Hassanpour GPA2 (2008) [51]	$(3.9444, 3.9444, 3.9444)$	$(0.4444, 0.4444, 0.4444)$	5.737	9.1423	0.8571	5.8340
Chen and Hsueh [46] (2007)	$(3.667, 3.945, 3.2230)$	$(0.444, 0.444, 0.444)$	6.246	9.979	0.6928	5.5640
Hassanpour GPA1 (2008) [51]	$(2.6154, 2.6154, 2.6154)$	$(0.6923, 0.6923, 0.6923)$	6.285	9.4093	0.7658	6.5769
Lu and Wang (2009) [20][3]	$(2.9524 + 0.5238x_i^c$ $, 2.3516\alpha_{x_i} - 0.6758))_L$		6.340	9.4920	0.5939	5.9452
Yeh (2009) [58]	$(3.530, 3.530, 3.530)$	$(0.5250, 0.5250, 0.5250)$	6.699	10.628	0.9169	5.8900
Arabpour and Tata method 1 (2008) [1][2]	$(3.4877, 3.4877, 3.4877)$	$(0.5306, 0.5306, 0.5306)$	6.739	10.687	0.9191	5.8968
Bargiela et al. (2007) [12]	$(3.4467, 3.4467, 3.4467)$	$(0.5360, 0.5360, 0.5360)$	6.776	10.742	0.9212	5.9033
Chen and Dang (2008) [54]	$(3.5284, 3.5284, 3.5284) + \widetilde{E}_i$	$(0.5298, 0.5298, 0.5298)$	7.000	10.808	0.8146	5.3868
Kao and Chyu (2002) [3]	$(3.3324, 3.5724, 3.8124)$	$(0.5193, 0.5193, 0.5193)$	7.470	12.030	0.8501	5.8175
Nasrabadi and Nasrabadi (2004) [35][4]	$(3.5767, 3.5767, 3.5767)$	$(-0.453, 0.5467, 1.5467)$	7.541	11.902	0.7948	5.9166
Chen and Hsueh [28] (2009)	$(3.2720, 3.5720, 3.8720)$	$(0.519, 0.519, 0.519)$	7.668	12.352	0.8375	5.8165
Arabpour and Tata method 2 (2008) [1]	$(3.2758, 3.5724, 3.8386)$	$(0.5188, 0.5193, 0.5235)$	7.691	12.382	0.8388	5.8208
Diamond (1988) [27][1]	$(3.2636, 3.5632, 3.8628)$	$(0.5206, 0.5206, 0.5206)$	7.709	12.379	0.8387	5.8221
Modarres et al. (RNP) (2004) [59][2]	$(3.278, 3.511, 3.744)$	$(0.542, 0.553, 0.562)$	7.783	12.448	0.8110	5.9337
Yang and Ku (2002) [57]	$(3.2052, 3.4967, 3.7882)$	$(0.5251, .5293, 0.5335)$	7.867	12.682	0.8429	5.8525
Hojati et al. (2005) [34][2]	$(3.00, 3.41, 3.82)$	$(0.50, 0.52, 0.54)$	8.284	13.464	0.7920	5.8200
Modarres et al. (RAP) (2004) [59][2]	$(3.152, 3.525, 3.898)$	$(0.485, 0.537, 0.589)$	9.169	14.995	0.7727	6.1200
Kao and Chyu (2003) [6]	$(2.603, 3.554, 4.503)$	$(0.522, 0.522, 0.522)$	9.363	15.966	0.8358	5.9575
Sakawa and Yano (1992) [24]	$(3.031, 3.201, 3.371)$	$(0.498, 0.579, 0.659)$	9.431	15.272	0.7717	6.1801

[1] The estimated models error of Diamond [27] is taken from [3], [54].
[2] The estimated models error of Arabpour and Tata [1], Modarres *et al.* [59] and Hojati *et al.* [34] are taken from [51].
[3] This model is represented in the form of $(y_i^c, \alpha_{y_i})_L$ which is equal to $(y_i^c - |\alpha_{y_i}|, y_i^c, y_i^c + |\alpha_{y_i}|)$.
[4] This model applied new addition and multiplication operators.

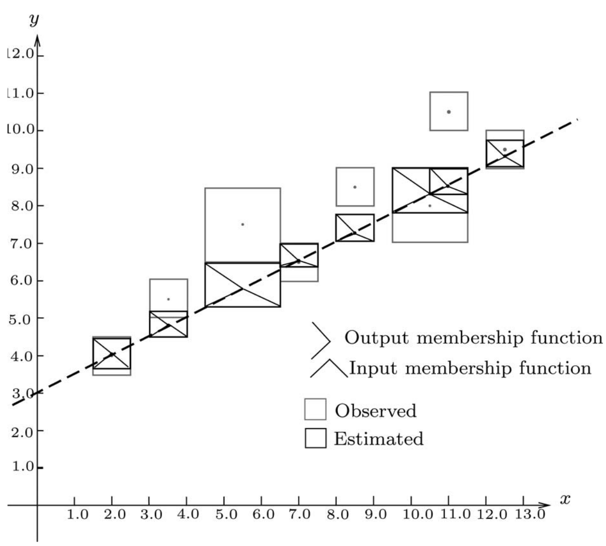

Fig. 8. Estimated fuzzy function for the given dataset in Table XIV.

other methods [1], [14], [56] are given in Table XVIII. The results show that our model is superior to the other models in the literature over NIA, RNIA, distance criterion, and Hojati's measure. While there are many methods in the literature, which are not capable of finding functions with negative coefficients (e.g., [3], [6], [24], [27], and [34]), our method is able to find a function with negative coefficients. Table XIII reports the performance of the proposed method for 1000 separate runs. Even the worst produced error (NIA) by the method is smaller when compared with the errors produced by other methods presented in the literature.

TABLE XVI
ARTIFICIAL ASYMMETRIC NUMERIC INPUT–FUZZY OUTPUT

Obs.	Independent variable	Dependent variable
1	1	$(7.0, 8.0, 15.0)$
2	2	$(8.10, 8.50, 10.80)$
3	3	$(8.20, 10.0, 10.20)$
4	4	$(8.40, 9.40, 11.0)$
5	5	$(8.50, 9.20, 11.20)$
6	6	$(7.50, 8.30, 12.0)$

TABLE XVII
EXPERIMENTAL RESULT FOR 1000 RUNS FOR THE DATASET GIVEN IN TABLE XVI BY THE PROPOSED METHOD

	best solution	worst solution	average	STD
Error (7)	7.10	9.02	7.60	0.32
Error (11)	5.20	6.45	5.54	0.21
Iterations	122	108	125	17

B. Symmetric Fuzzy Input–Fuzzy Output Results

This class involves a more generalized form of numeric input–fuzzy output. Sakawa and Yano [24] introduced a fuzzy input–fuzzy output dataset, which is given in Table XIV.

Many approaches present in the literature (in [3], [6], [12], [20], [24], [27], [34], [54], and [57]) have used this dataset for validation purposes. The result of the comparison between our estimated model and other methods' model is given Table XV. The comparison results show that the model generates a smaller error RNIA than reported for other models generated by other techniques. However, based on the FR model comparison conditions, our model is superior to Hassanpour's model [51] and Pareto-equivalent to the rest of the models over RNIA, NIA, distance criterion, and Hojati's measure. Fig. 8 illustrates the behavior of the estimated fuzzy linear model for the dataset that is given in Table XIV. The dashed and continuous lines show the

TABLE XVIII
ESTIMATED MODELS AND THEIR ERRORS FOR THE DATASET GIVEN IN TABLE XII

Method	A_0	A_1	NIA	RNIA	T_e (Hojati's criterion)	T_e (Distance
This work	$(-5.85, -5.83, -5.82)$	$(1.21, 1.24, 1.33)$	6.89	10.29	0.7056	9.14
Arabpour and Tata (2008) [1]	$(-5.25, -4.90, -4.54)$	$(1.14, 1.20, 1.25)$	10.3944	16.63	0.8446	9.98
Yao and Yu (2006) [56]	$(-7.88, -4.89, -3.29)$	$(1.20, 1.20, 1.120)$	19.7039	31.24	0.8027	13.29
Hong and Hwang (2003) [14]	$(-5.45, -5.45, -5.45)$	$(0.71, 1.22, 1.73)$	34.8130	42.42	0.8250	19.15

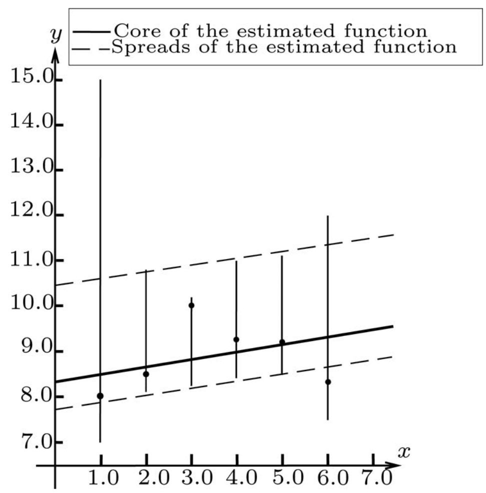

Fig. 9. Estimated fuzzy function for the dataset given in Table XVI.

estimated fuzzy output and the observed outputs, respectively. According to the reported results in Table XIX, the average error RNIA for 1000 separate runs is lower in comparison with the results produced by other models.

C. Numeric Input–Asymmetric Fuzzy Output Results

To show the performance of the proposed method for numeric input–asymmetric fuzzy output, we have designed an artificial benchmark (see Table XVI). Although there is a numeric input–asymmetric fuzzy output in the literature for fuzzy multiple linear regression [47], the reason to introduce a new dataset is to investigate the performance of the approach for a dataset with very large differences in the spreads of each datum. If the difference in the spreads is not noticeable, then even the approaches that are only capable of having symmetric coefficients (such as [34]) may result in almost the same error level as the ones with asymmetric coefficients. After applying the method to this dataset, the best model comes in the form (14). The total errors for the obtained model in (14) with NIA, Hojati's measurement, and distance criterion are 5.20, 0.3364, and 3.6701, respectively

$$y = (0.1561, 0.1630, 0.1634)x_1 + (7.72, 8.33, 10.44). \quad (14)$$

As the method was run for 1000 times, the relevant statistics are reported in Table XVII. The plot of the best model given by (14) is included in Fig. 9. It has been found that the method can deal with asymmetric fuzzy data effectively. It is quite expected that as five (out of six) points of the dataset have larger right spreads than the left spreads, and the model follows this reflected tendency.

TABLE XIX
EXPERIMENTAL RESULT FOR 1000 RUNS FOR THE DATASET GIVEN IN TABLE XIV

	best solution	worst solution	average	STD
Error (7)	7.6404	9.3234	7.8158	0.1913
Error (11)	5.2656	6.1336	5.4919	0.1582
Iterations	229	227	187	25

D. Asymmetric Fuzzy Input–Asymmetric Fuzzy Output Results With Outliers

To test the performance of the proposed method for a large dataset and outliers, we applied the same approach and function as given by Buckley and Feuring [60]. The linear function is given as $\widetilde{y} = (5.0, 5.2, 5.7)\widetilde{x} + (3.0, 3.2, 3.7) + \epsilon$, where in 95% of the time, $\epsilon = \mathrm{RND}[-0.5, 0.5]$, and 5% of the time, $\epsilon = \mathrm{RND}[-5, -2]$ or $\mathrm{RND}[2, 5]$ is used to generate outputs from 1000 randomly chosen $\widetilde{x}$ in [11 000]. Then, we attempt to fit $\widetilde{y} = \widetilde{k}_1\widetilde{x} + \widetilde{k}_2$ to the generated asymmetric fuzzy input–asymmetric fuzzy output with outliers. Note that the dataset is generated once; however, we have run the method 1000 times. The average errors of RNIA, NIA, Hojati's measurement, and distance criterion for the best obtained model with $\widetilde{k}_1 = (5.0095, 5.4883, 5.6993)$ and $\widetilde{k}_2 = (2.8665, 3.0629, 4.0537)$ are 38.1736, 5.9010e+003, 0.0355, and 4.3985e+003, respectively. The average RNIA, CPU time, and standard deviations of RNIA and CPU time are 253.73, 66.4810 s, and 116.7378 and 6.48 s, respectively. For the best obtained model, the NIA between the original output and the ideal output and the estimated output and the ideal output for the outliers are represented by outliers' error and estimated error in Table XX. The ideal outputs are obtained from $\widetilde{y} = (5.5, 5.5, 5.7)\widetilde{x} + (3.0, 3.2, 3.7)$ with no noise. In Table XX, 25 out of 42 (59%) of the estimated outputs generate less error than the outliers' error. In other words, 59% of the outliers are detected, and the obtained model is not fitting them. However, one may design and apply a specific objective function to be minimized for datasets with outliers. As mentioned earlier, in this paper, we used the objective function given in (7) that is not necessarily the best objective function for the detection of outliers.

V. CONCLUSION

We have approached the solution of FLR problem with a global continuous optimization method. This method is the result of the hybridization of tabu search and improved harmony search. Tabu search is applied in the first phase, and it searches the near optimum solution, and then, the improved harmony search intensifies the area close to the near optimum solution.

TABLE XX
ESTIMATED ERROR (11) OF THE BEST OBTAINED MODEL FOR THE OUTLIERS IN THE LARGE DATASET

No.	Instance	Input	Ideal output	Outlier's output	Estimated by the model	Outliers' error	Estimated error
1	50	(577.9,580.6,583.2)	(2892.7,3022.4,3328.0)	(2900.9,3205.1,3336.2)	(2898.0,3189.7,3328.0)	136.2	123.1
2	55	(127.6,129.5,131.1)	(640.7,676.7,751.0)	(646.8,722.0,757.1)	(641.8,713.9,751.2)	37.7	28.4
3	81	(789.0,789.3,791.2)	(3948.1,4107.4,4513.4)	(3938.6,4335.0,4503.9)	(3955.4,4334.8,4513.2)	154.3	165.1
4	103	(222.0,222.1,224.2)	(1113.2,1158.2,1281.7)	(1104.7,1216.6,1273.2)	(1115.2,1222.1,1281.9)	37.6	47.3
5	108	(425.2,425.3,427.6)	(2128.9,2214.8,2440.9)	(2120.8,2334.5,2432.7)	(2132.8,2337.3,2441.0)	80.0	89.6
6	116	(118.6,121.2,121.4)	(595.8,633.5,695.9)	(588.1,662.4,688.2)	(596.8,668.3,696.1)	18.4	26.4
7	181	(9.8,11.4,11.8)	(51.8,62.3,71.2)	(58.8,73.0,78.2)	(51.8,65.5,71.5)	12.7	2.9
8	184	(873.4,875.6,876.7)	(4369.8,4556.3,5001.1)	(4361.1,4810.5,4992.3)	(4378.0,4808.5,5000.8)	173.8	183.5
9	204	(273.1,273.9,276.5)	(1368.3,1427.5,1579.7)	(1362.1,1503.8,1573.5)	(1370.8,1506.3,1579.8)	51.2	58.5
10	220	(820.8,823.6,826.1)	(4107.0,4286.0,4712.7)	(4097.5,4523.9,4703.2)	(4114.7,4523.3,4712.5)	162.8	173.6
11	226	(482.9,485.5,485.6)	(2417.6,2527.8,2771.7)	(2410.5,2666.5,2764.5)	(2422.1,2667.6,2771.7)	93.8	102.1
12	242	(463.5,466.3,469.0)	(2320.5,2428.0,2676.9)	(2330.2,2577.9,2686.6)	(2324.7,2562.3,2676.9)	114.4	99.4
13	246	(494.1,495.3,497.8)	(2473.6,2578.8,2840.9)	(2465.5,2719.5,2832.8)	(2478.2,2721.5,2840.9)	95.1	104.7
14	272	(88.2,91.0,92.7)	(444.1,476.2,532.2)	(437.9,497.6,526.0)	(444.8,502.3,532.5)	14.0	20.6
15	278	(734.2,736.6,739.5)	(3673.8,3833.3,4218.8)	(3679.5,4060.2,4224.4)	(3680.7,4045.5,218.6)	165.4	155.5
16	280	(288.6,290.9,293.4)	(1446.0,1515.8,1676.2)	(1439.8,1597.2,1670.0)	(1448.6,1599.6,1676.4)	55.2	62.6
17	295	(408.4,410.0,410.7)	(2045.2,2135.2,2344.5)	(2054.7,2268.1,2354.1)	(2048.9,2253.2,2344.6)	100.7	86.3
18	315	(957.6,958.0,960.3)	(4791.1,4985.0,5477.2)	(4785.8,5267.5,5471.9)	(4800.0,5261.1,5476.9)	195.5	200.4
19	323	(38.6,41.2,44.0)	(195.9,217.6,254.5)	(190.4,224.7,249.0)	(196.1,229.3,254.8)	5.7	10.1
20	350	(56.8,58.5,60.6)	(286.8,307.5,348.8)	(293.2,331.8,355.3)	(287.2,324.3,349.1)	23.48	13.5
21	407	(882.4,884.3,885.2)	(4415.1,4601.5,5049.4)	(4420.2,4872.1,5054.4)	(4423.3,4856.2,5049.1)	194.3	1185.1
22	507	(930.0,932.2,933.7)	(4652.9,4850.6,5326.0)	(4646.1,5123.7,5319.1)	(4661.6,5119.2,5325.7)	188.3	195.5
23	561	(6.5,7.0,7.8)	(35.5,39.8,47.9)	(44.8,51.5,57.1)	(35.5,41.7,48.2)	11.7	1.8
24	604	(333.0,333.1,336.7)	(1667.9,1738.8,1922.6)	(1660.2,1831.6,1915.0)	(1670.9,1834.9,1922.7)	62.1	71.1
25	629	(391.4,393.8,396.3)	(1960.0,2051.2,2262.8)	(1968.4,2178.0,2271.2)	(1963.6,2164.6,2262.9)	97.1	84.1
26	653	(662.3,664.2,667.2)	(3314.6,3457.0,3806.5)	(3323.7,3665.7,3815.6)	(3320.8,3648.4,3806.4)	154.8	140.3
27	675	(15.3,17.7,18.5)	(79.3,95.3,109.0)	(74.8,96.5,104.5)	(79.3,100.3,109.4)	3.8	4.4
28	716	(66.1,67.8,68.6)	(333.0,355.7,394.7)	(325.1,368.4,386.8)	(333.5,375.1,395.0)	8.9	15.1
29	763	(730.4,733.2,735.8)	(3654.9,3816.0,4197.7)	(3645.0,4026.4,4187.8)	(3661.7,4027.3,4197.6)	143.4	154.8
30	770	(294.1,296.5,297.0)	(1473.6,1544.9,1696.9)	(1467.3,1627.7,1690.5)	(1476.3,1630.2,1697.0)	55.4	62.9
31	775	(331.0,331.9,333.3)	(1657.7,1728.8,1903.6)	(1653.0,1824.0,1898.9)	(1660.7,1824.4,1903.7)	64.7	70.1
32	787	(696.0,696.0,698.9)	(3483.2,3622.7,3987.7)	(3489.5,3838.1,3994.0)	(3489.7,3823.2,3987.6)	156.7	146.1
33	797	(542.3,544.3,545.2)	(2714.5,2833.4,3111.5)	(2705.9,2988.5,3102.9)	(2719.5,2990.2,3111.4)	104.5	114.5
34	820	(964.3,966.4,968.6)	(4824.3,5028.4,5524.7)	(4833.3,5327.6,5533.7)	(4833.3,5306.8,5524.3)	217.9	202.8
35	836	(199.8,202.1,203.0)	(1001.9,1054.1,1161.1)	(1007.5,1120.6,1166.7)	(1003.7,1112.2,1161.3)	52.0	43.4
36	871	(31.4,33.3,35.5)	(159.8,176.6,205.9)	(150.3,177.4,196.5)	(160.0,186.0,206.3)	8.8	8.1
37	875	(224.8,226.5,227.9)	(1127.0,1181.1,1302.9)	(1120.5,1242.9,1296.4)	(1129.0,1246.3,1303.1)	40.9	48.5
38	898	(654.0,654.6,655.5)	(3273.1,3406.9,3740.0)	(3279.6,3609.9,3746.4)	(3279.2,3595.4,3739.9)	147.3	136.8
39	899	(616.6,619.5,620.4)	(3085.8,3224.6,3539.8)	(3079.9,3404.9,3533.9)	(3091.5,3403.1,3539.7)	124.0	130.5
40	955	(516.7,519.4,521.4)	(2586.7,2704.2,2975.5)	(2578.7,2852.3,2967.5)	(2591.4,2853.8,2975.4)	100.7	110.0
41	963	(730.1,730.5,732.4)	(3653.6,3801.9,4178.6)	(3661.6,4029.3,4186.6)	(3660.4,4012.4,4178.5)	165.9	153.0
42	985	(531.5,533.8,535.4)	(2660.5,2779.1,3055.5)	(2651.1,2930.2,3046.1)	(2665.4,2932.9,3055.5)	101.7	112.7

TABLE XXI
INCORRECT REPORTED FUZZY OUTPUT GIVEN BY SHAKOURI AND NADIMI IN [8] FOR DATASET GIVEN IN TABLE VI VERSUS THE ACTUAL FUZZY OUTPUTS

Obs.	Independent variable	Reported Output [8]	Actual output	Actual error
Team 1	$(2.27, 5.83, 9.39)$	$(4.98, 7.18, 9.38)$	$(4.77, 7.20, 9.63)$	2.43 (0.69)
Team 2	$(0.33, 0.85, 1.37)$	$(0.40, 0.97, 1.54)$	$(-2.72, 1.00, 4.71)$	3.20 (6.16)
Team 3	$(5.43, 13.93, 22.43)$	$(2.33, 3.94, 5.55)$	$(1.29, 3.96, 6.64)$	11.05 (1.30)
Team 4	$(1.56, 4, 6.44)$	$(1.57, 3.12, 4.67)$	$(0.56, 3.15, 5.73)$	1.56 (0.64)
Team 5	$(0.64, 1.65, 2.66)$	$(1.96, 2.87, 3.77)$	$(-0.67, 2.89, 6.46)$	2.82 (2.79)
Team 6	$(0.62, 1.58, 2.54)$	$(0.62, 1.59, 2.57)$	$(-1.58, 1.62, 4.82)$	2.24 (2.33)
Team 7	$(3.19, 8.18, 13.17)$	$(2.04, 2.65, 3.26)$	$(-1.34, 2.68, 6.70)$	7.64 (1.53)
Team 8	$(0.72, 1.85, 2.98)$	$(0.72, 1.85, 2.98)$	$(-1.09, 1.88, 4.84)$	1.83 (1.62)
T_e (RNIA (NIA))	-	(5.16)	32.78 (17.06)	32.78 (17.06)

The main benefit to apply this method over the existing methods is that no assumptions, such as differentiability or continuity, are required. In light of the experimental results, our method is either superior or Pareto-equivalent to the other methods discussed in the literature for all the given benchmark datasets. Moreover, the performance of the method is validated for a large dataset in the presence of outliers.

Considering these encouraging results, our intention is to apply the same idea to nonlinear and nonparametric regression. In the nonlinear case, if the form of the function is known, then only the number of variables increases for the optimization process. However, for nonparametric regression, a potential approach is to apply tabu programming to find the form of the function first. Then, an intensifier can be applied to tune the coefficients of the function.

APPENDIX

The reported function in [8] is $\widetilde{y} = (-20.08, -20.08, -20.08) + (-0.23, -0.16, -0.09)x_1 + (-1.22, -0.9, -0.58) x_2 + (1.66, 1.81, 1.96)x_3$. Applying the fuzzy arithmetic for addition and multiplication of the fuzzy numbers [61], the results are given in Table XXI.

Acknowledgment

The authors would like to thank the four anonymous reviewers for their constructive comments and suggestions. They are also grateful to Prof N. Kasabov for fruitful discussions and R. Sepahi for his help in typesetting of this paper.

References

[1] A. R. Arabpour and M. Tata, "Estimating the parameters of a fuzzy linear regression model," *Iranian J. Fuzzy Syst.*, vol. 5, no. 2, pp. 1–19, 2008.
[2] P. D. Urso and A. Santoroa, "Fuzzy clusterwise linear regression analysis with symmetrical fuzzy output variable," *Comput. Statist. Data Anal.*, vol. 51, no. 1, pp. 287–313, 2006.
[3] C. Kao and C.-L. Chyu, "A fuzzy linear regression model with better explanatory power," *Fuzzy Sets Syst.*, vol. 126, no. 3, pp. 401–409, 2002.
[4] T. Feuring, W. Golubski, and M. Gassmann, "Fuzzy regression: A genetic programming approach," in *Proc. 4th Int. Conf. Knowl.-Based Intell. Eng. Syst. All. Technol.*, Brighton, U.K., 2000, pp. 349–352.
[5] R. M. Dom, "Fuzzy regression model for the prediction of oral cancer susceptibility," Ph.D. dissertation, Univ. Malaya, Kuala Lumpur, Malaysia, 2009.
[6] C. Kao and C.-L. Chyu, "Least-squares estimates in fuzzy regression analysis," *Eur. J. Oper. Res.*, vol. 148, no. 2, pp. 426–435, 2003.
[7] Y.-H. O. Chang and B. M. Ayyub, "Fuzzy regression methods: A comparative assessment," *Fuzzy Sets Syst.*, vol. 119, no. 2, pp. 187–203, 2001.
[8] H. Shakouri and R. Nadimi, "A novel fuzzy linear regression model based on a non-equality possibility index and optimum uncertainty," *Appl. Soft Comput.*, vol. 9, no. 2, pp. 590–598, 2009.
[9] M.-S. Yang and C.-H. Ko, "On cluster-wise fuzzy regression analysis," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, no. 1, pp. 1–13, Feb. 1997.
[10] B. Kim and R. R. Bishu, "Evaluation of fuzzy linear regression models by comparing membership functions," *Fuzzy Sets Syst.*, vol. 100, no. 1–3, pp. 343–352, 1998.
[11] H. Tanaka, S. Uejima, and A. Asal, "Linear regression analysis with fuzzy model," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-12, no. 6, pp. 903–907, Nov. 1982.
[12] A. Bargiela, W. Pedrycz, and T. Nakashima, "Multiple regression with fuzzy data," *Fuzzy Sets Syst.*, vol. 158, no. 19, pp. 2169–2188, 2007.
[13] R. A. Aliev, B. Fazlollahi, and R. M. Vahidov, "Genetic algorithms-based fuzzy regression analysis," *Soft Comput.*, vol. 6, no. 6, pp. 470–475, 2002.
[14] D. H. Hong and C. Hwang, "Support vector fuzzy regression machines," *Fuzzy Sets Syst.*, vol. 138, no. 2, pp. 271–281, 2003.
[15] D. T. Redden and W. H. Woodall, "Further examination of fuzzy linear regression," *Fuzzy Sets Syst.*, vol. 79, pp. 203–211, 1996.
[16] W. Näther, "Regression with fuzzy random data," *Comput. Statist. Data Anal.*, vol. 51, no. 1, pp. 235–252, 2006.
[17] P.-Y. Hao and J.-H. Chiang, "Fuzzy regression analysis by support vector learning approach," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 2, pp. 428–441, Apr. 2008.
[18] D. H. Hong and C. Hwang, "Interval regression analysis using quadratic loss support vector machine," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 2, pp. 229–237, Apr. 2005.
[19] R. Coppi, "Management of uncertainty in statistical reasoning: The case of regression analysis," *Int. J. Approx. Reason.*, vol. 47, no. 3, pp. 284–305, 2008.
[20] J. Lu and R. Wang, "An enhanced fuzzy linear regression model with more flexible spreads," *Fuzzy Sets Syst.*, vol. 160, no. 17, pp. 2505–2523, 2009.
[21] J. J. Buckley, "Fuzzy prediction in linear regression," *Stud. Fuzz. Soft Comput.*, vol. 196, pp. 177–179, 2006.
[22] S. Imoto, Y. Yabuuchi, and J. Watada, "Fuzzy regression model of R&D project evaluation," *Appl. Soft Comput.*, vol. 8, no. 3, pp. 1266–1273, 2008.
[23] J. A. Sánchez, "Calculating insurance claim reserves with fuzzy regression," *Fuzzy Sets Syst.*, vol. 157, no. 23, pp. 3091–3108, 2006.
[24] M. Sakawa and H. Yano, *Fuzzy Linear Regression and its Applications*, Fuzzy Regression Analysis.. Heidelberg, Germany: Physica-Verlag, 1992, pp. 61–80.
[25] S. M. Taheri, "Trends in fuzzy statistics," *Austrian J. Statist.*, vol. 32, no. 3, pp. 239–257, 2003.
[26] A. Celmins, "Least squares model fitting to fuzzy vector data," *Fuzzy Sets Syst.*, vol. 22, no. 3, pp. 245–269, 1987.
[27] P. Diamond, "Fuzzy least squares," *Inf. Sci.*, vol. 46, no. 3, pp. 141–157, 1988.
[28] L.-H. Chen and C.-C. Hsueh, "Fuzzy regression models using the least-squares method based on the concept of distance," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 6, pp. 1259–1272, Dec. 2009.
[29] C. Li, X. Liao, and J. Yu, "Tabu search for fuzzy optimization and applications," *Inf. Sci.*, vol. 158, pp. 3–13, 2004.
[30] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 36–38, pp. 3902–3933, 2005.
[31] R. Chelouah and P. Siarry, "A hybrid method combining continuous tabu search and nelder-mead simplex algorithms for the global optimization of multiminima functions," *Eur. J. Oper. Res.*, vol. 161, no. 3, pp. 636–654, 2005.
[32] M. H. Mashinchi, M. A. Orgun, and W. Pedrycz, "Hybrid optimization with improved tabu search," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1993–2006, 2010.
[33] A.-R. Hedar and M. Fukushima, "Tabu search by direct search methods for nonlinear global optimization," *Eur. J. Oper. Res.*, vol. 170, no. 2, pp. 329–349, 2006.
[34] M. Hojati, C. R. Bector, and K. Smimou, "A simple method for computation of fuzzy linear regression," *Eur. J. Oper. Res.*, vol. 166, no. 1, pp. 172–184, 2005.
[35] M. M. Nasrabadi and E. Nasrabadi, "A mathematical-programming approach to fuzzy linear regression analysis," *Appl. Math. Comput.*, vol. 55, no. 3, pp. 873–881, 2004.
[36] M. Fesanghary, M. Mahdavi, M. M Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems," *Comput. Methods Appl. Mech. Eng.*, vol. 197, no. 33–40, pp. 3080–3091, 2008.
[37] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Appl. Math. Comput.*, vol. 198, no. 2, pp. 643–656, 2008.
[38] M. H. Mashinchi, M. R. Mashinchi, and M. Mashinchi, "A two-phased genetic algorithm approach for linear and quadratic fuzzy regression," *Set-valued Math. Appl.*, vol. 1, no. 2, pp. 183–190, 2008.
[39] K. Y. Chan, C. K. Kwong, and Y. C. Tsim, "A genetic programming based fuzzy regression approach to modelling manufacturing processes," *Int. J. Prod. Res.*, vol. 48, no. 7, pp. 1967–1982, 2009.
[40] M. H. Mashinchi, M. A. Orgun, and M. Mashinchi, "Solving fuzzy linear regression with hybrid optimization," in *Neural Information Processing*, (Lecture Notes in Computer Science 5864), C.-S. Leung, M. Lee, and J. H. Chan, Eds. New York: Springer-Verlag, 2009, pp. 336–343.
[41] N. Kasabov, "Evolving intelligence in humans and machines: Integrative evolving connectionist systems approach," *IEEE Comput. Intell. Mag.*, vol. 3, no. 3, pp. 23–37, Aug. 2008.
[42] M. R. Mashinchi and A. Selamat, "An improvement on genetic-based learning method for fuzzy artificial neural networks," *Appl. Soft Comput.*, vol. 9, no. 4, pp. 1208–1216, 2009.
[43] M. J. Watts, "A decade of Kasabovs evolving connectionist systems: A review," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 39, no. 3, pp. 253–269, May 2009.
[44] A. Abdalla and J. J. Buckley, "Monte Carlo methods in fuzzy linear regression II," *Soft Comput.*, vol. 12, no. 5, pp. 463–468, 2008.
[45] M. Modarres, E. Nasrabadi, and M. M. Nasrabadi, "Fuzzy linear regression models with least square errors," *Appl. Math. Comput.*, vol. 163, no. 2, pp. 977–989, 2005.
[46] L.-H. Chen and C.-C. Hsueh, "A mathematical programming method for formulating a fuzzy regression model based on distance criterion," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 37, no. 3, pp. 705–712, Jun. 2007.
[47] P. D. Urso, "Linear regression analysis for fuzzy/crisp input and fuzzy/crisp output data," *Comput. Statist. Data Anal.*, vol. 42, no. 1–2, pp. 47–72, 2003.
[48] A. Bisserier, R. Boukezzoula, and S. Galichet, "A revisited approach to linear fuzzy regression using trapezoidal fuzzy intervals," *Inf. Sci.*, vol. 180, no. 19, pp. 3653–3673, 2010.
[49] R. T. Zheng, N. Q. Ngo, P. Shum, S. C. Tjin, and L. N. Binh, "A staged continuous tabu search algorithm for the global optimization and its applications to the design of fiber bragg gratings," *Comput. Optim. Appl.*, vol. 30, pp. 319–335, 2005.
[50] D. Savic and W. Pedrycz, "Evolution of fuzzy linear regression models," *Fuzzy Sets Syst.*, vol. 39, pp. 51–63, 1991.
[51] H. Hassanpour, "A goal programming approach to fuzzy linear regression," Ph.D. dissertation, Shahid Bahonar Univ. Kerman , Iran, 2008.

[52] H. Tanaka, H. Ishibuchi, and J. Watada, "Probablistic linear regression for analysis for fuzzy data," *Eur. J. Oper. Res.*, vol. 40, pp. 389–396, 1989.

[53] E. C. Özelkan and L. Duckstein, "Multi-objective fuzzy regression: A general framework," *Comput. Oper. Res.*, vol. 27, no. 7–8, pp. 635–652, 2000.

[54] S.-P. Chen and J.-F. Dang, "A variable spread fuzzy linear regression model with higher explanatory power and forecasting accuracy," *Inf. Sci.*, vol. 178, no. 20, pp. 3973–3988, 2008.

[55] C. Kao and P.-H. Lin, "Entropy for fuzzy regression analysis," *Int. J. Syst. Sci.*, vol. 36, no. 14, pp. 869–876, 2005.

[56] C.-C. Yao and P.-T. Yu, "Fuzzy regression based on asymmetric support vector machines," *Appl. Math. Comput.*, vol. 182, no. 1, pp. 175–193, 2006.

[57] M.-S. Yang and T.-S. Lin, "Fuzzy least-squares linear regression analysis for fuzzy input-output data," *Fuzzy Sets Syst.*, vol. 126, no. 3, pp. 389–399, 2002.

[58] C.-T. Yeh, "Reduction to least-squares estimates in multiple fuzzy regression analysis," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 935–948, Aug. 2009.

[59] M. Modarres, E. Nasrabadi, and M. M. Nasrabadi, "Fuzzy linear regression analysis from the point of view risk," *Int. J. Uncertainty, Fuzz. Knowl.-Based Syst.*, vol. 12, no. 5, pp. 635–649, 2004.

[60] J. J. Buckley and T. Feuring, "Linear and non-linear fuzzy regression: evolutionary algorithm solutions," *Fuzzy Sets Syst.*, vol. 112, no. 3, pp. 381–394, 2000.

[61] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

M. Hadi Mashinchi received the M.Sc. degree in computer science from University Technology Malaysia, Kuala Lumpur, Malaysia in 2007. He is currently working toward the Ph.D. degree with Macquarie University, Sydney, Australia.

He is currently a Data Mining Consultant in Sydney. His research interests include hybrid optimization, global continuous optimization, fuzzy regression, and the application of data mining in engineering applications. He has served on the review boards of journals and international conferences. He is also with the Program Committee of the 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, to be held in Moscow, Russia.

Mehmet A. Orgun (SM'96) received the B.Sc. and M.Sc. degrees in computer science and engineering from Hacettepe University, Ankara, Turkey, in 1982 and 1985, respectively, and the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada, in 1991.

He is currently a Professor with Macquarie University, Sydney, Australia. Prior to joining Macquarie University in September 1992, he worked as a Postdoctoral Fellow with the University of Victoria. His research is in the broad area of intelligent systems, with specific research interests in knowledge discovery, multiagent systems, trusted systems, and temporal reasoning. His professional service includes editorial/review board memberships of leading journals, as well as program committee or senior program committee memberships with numerous national and international conferences. He was recently the Program Committee co-chair of the 20th Australian Joint Conference on Artificial Intelligence and the 14th Pacific-Rim International Conference on Artificial Intelligence, as well as the Conference co-chair of the Second and Third International Conferences on Security of Information and Networks.

Mashaallah Mashinchi received the B.Sc. degree from the Ferdowsi University of Mashhad, Mashhad, Iran, the M.Sc. degree from Shiraz University, Shiraz, Iran, both in statistics, in 1976 and 1978, respectively, and the Ph.D. degree in mathematics from Waseda University, Tokyo, Japan, in 1987.

He is currently a Professor with the Department of Statistics, Shahid Bahonar University of Kerman (SBUK), Kerman, Iran. He is the Editor-in-Chief of the *Iranian Journal of Fuzzy Systems*, and the director of the Fuzzy Systems Applications Center of Excellence at SUBK. His research interests include fuzzy mathematics, especially on statistics, decision making, and algebraic systems.

Witold Pedrycz (M'88–SM'90–F'99) is a Professor and Canada Research Chair (CRC–Computational Intelligence) with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. He also holds an appointment of special professorship with the School of Computer Science, University of Nottingham,Nottingham, U.K. In 2009, he was elected a foreign member of the Polish Academy of Sciences. His main research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He has published numerous papers in these areas. He is also the author of 14 research monographs covering various aspects of Computational Intelligence and Software Engineering. He has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He is intensively involved in editorial activities. He is an Editor-in-Chief of *Information Sciences* and Editor-in-Chief of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A. He currently serves as an Associate Editor of IEEE TRANSACTIONS ON FUZZY SYSTEMS and a number of other international journals. He has edited a number of volumes; the most recent one is entitled *Handbook of Granular Computing* (Wiley, 2008).

Dr. Pedrycz received the prestigious Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Council in 2007. He received the 2008 IEEE Canada Computer Engineering Medal. In 2009, he received a Cajastur Prize for Soft Computing from the European Centre for Soft Computing for "pioneering and multifaceted contributions to Granular Computing."