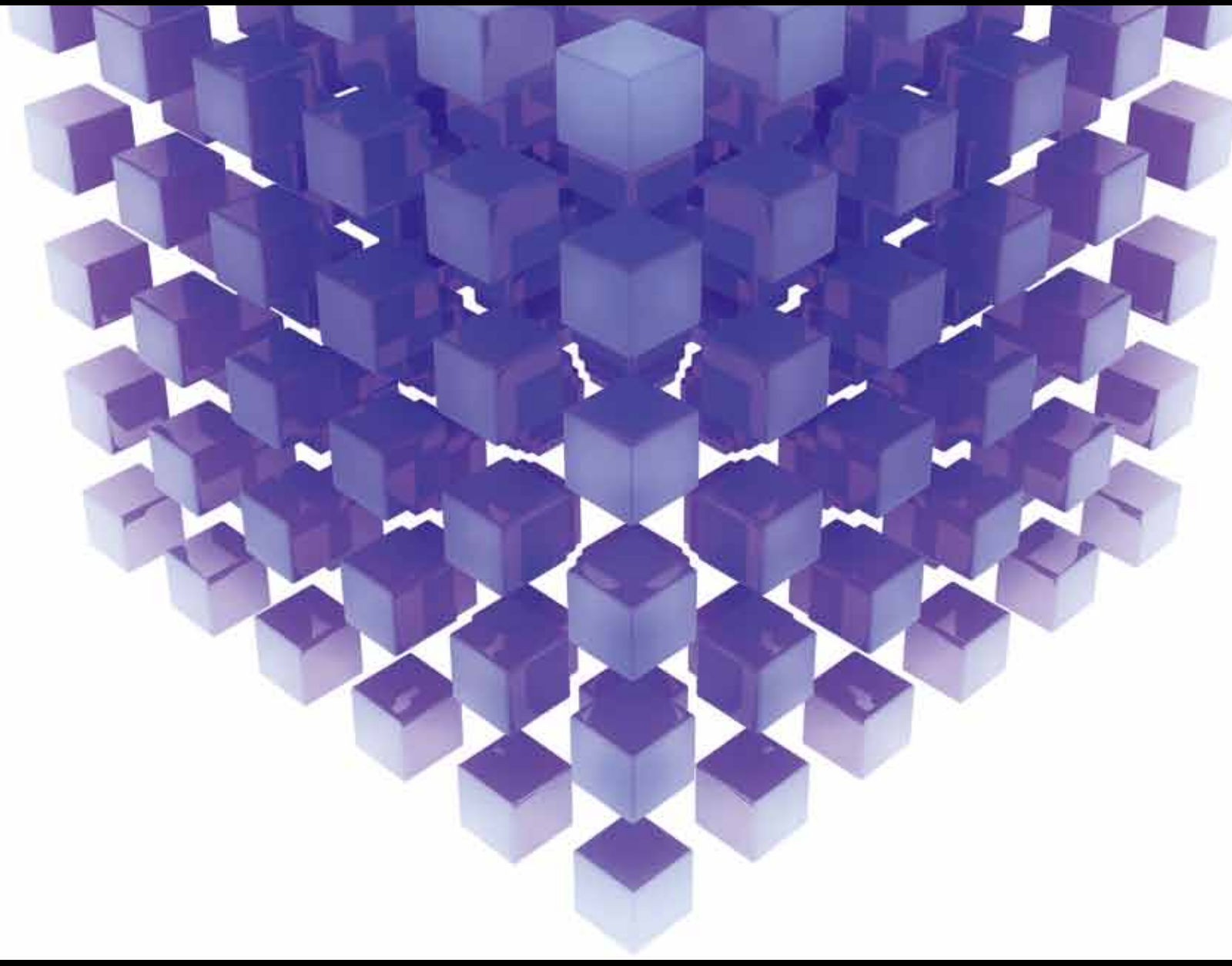
Mathematical Problems in Engineering

Applications of Fuzzy Ensemble Approaches in Modeling, Forecasting, and Control

Guest Editors: Toly Chen, Katsuhiro Honda, Pedro Ponce, T. Warren Liao, and Yi-Chi Wang



Applications of Fuzzy Ensemble Approaches in Modeling, Forecasting, and Control

Mathematical Problems in Engineering

Applications of Fuzzy Ensemble Approaches in Modeling, Forecasting, and Control

Guest Editors: Toly Chen, Katsuhiro Honda, Pedro Ponce, T. Warren Liao, and Yi-Chi Wang

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 607254, 2 pages
http://dx.doi.org/10.1155/2013/607254

Editorial

Applications of Fuzzy Ensemble Approaches in Modeling, Forecasting, and Control

Toly Chen,[1] Katsuhiro Honda,[2] Pedro Ponce,[3] T. Warren Liao,[4] and Yi-Chi Wang[1]

[1] *Department of Industrial Engineering and Systems Management, Feng Chia University, Taichung 407, Taiwan*
[2] *Graduate School of Engineering, Osaka Prefecture University, Osaka 599-8531, Japan*
[3] *Posgrados en Ciencias de la Ingeniería, TEC de Monterrey, Campus Ciudad de México, Tlalpan C.P. 14380, Mexico*
[4] *Department of Mechanical and Industrial Engineering, Louisiana State University, Baton Rouge, LA 70803, USA*

Correspondence should be addressed to Toly Chen; tcchen@fcu.edu.tw

Received 20 November 2013; Accepted 20 November 2013

Fuzzy systems have been successfully applied to various fields. However, most applications are based on a single fuzzy system. The simultaneous use of multiple fuzzy systems may be able to achieve better performances. These systems, homogeneous or heterogeneous, collaborate to enhance the effectiveness or efficiency of solving a problem. Based on this point of view, the ensemble of fuzzy systems has been widely used for the modeling, forecasting, and control of complex systems under an uncertain, restricted, or subjective environment. The purpose of this special issue is to provide the details of recent advances in developing fuzzy ensemble approaches and their applications. The target audiences are researchers in engineering mathematics, information engineering, information management, artificial intelligence, and computational intelligence, as well as practicing managers/engineers. After a very strict review process, only five papers from researchers around the world were finally accepted. A brief summary of each is described below.

M.-Y. Shan et al. proposed a combination of cloud model and quantum-behaved particle swarm optimization (COC-QPSO) to solve the problems of fitting complex functions and clustering various datasets. They also investigated the relation between Gaussian and fat-tail distributions, so as to construct a fat-tail distribution from Gaussian distribution in an iterative way to depict more uncertain phenomena.

H.-C. Wu and T. Chen proposed a fuzzy-neural ensemble and geometric rule fusion approach to optimize the performance of job dispatching in a wafer fabrication factory. Two dispatching rules were fused, and job slacks were diversified by maximizing the geometric mean of the neighboring distances of slacks. In addition, the fuzzy c-means (FCM) and back propagation network (BPN) ensemble approach was also applied to estimate the remaining cycle time of a job, which then served as an input to the new rule.

The existing nonlinear programming (NLP) based fuzzy collaborative intelligence methods cannot distinguish between the different expert opinions. It is also not easy to find the global optimal solution using these methods. In order to solve these problems and to improve the performance of the semiconductor yield forecasting, T. Chen and Y.-C. Wang proposed a quadratic programming (QP) based fuzzy collaborative intelligence approach.

H. Ponce et al. presented a new fuzzy inference model based on artificial hydrocarbon networks. In addition, the fuzzy-molecular inference model (FIM-model) was also proposed to partition the output space in the defuzzification step according to the information from the molecular units. Such molecules are linguistic units that can be partially understandable through the organized structure of the topology and the metadata parameters involved in the artificial hydrocarbon network.

X. Liu and Y. Chen proposed a systematic approach to optimize the h value in a fuzzy linear regression (FLR) problem, based on the minimum fuzziness criteria with symmetric triangular fuzzy numbers (TFNs). They defined a new concept of credibility to evaluate the performance of

an FLR model with different h values for a set of sample data pairs. Based on this concept, a mathematical programming model was formulated and solved to optimize the h value.

Acknowledgment

We sincerely thank all the authors and reviewers for their valuable contributions to this special issue.

Toly Chen
Katsuhiro Honda
Pedro Ponce
T. Warren Liao
Yi-Chi Wang

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 210164, 9 pages
http://dx.doi.org/10.1155/2013/210164

Research Article

A Systematic Approach to Optimizing h Value for Fuzzy Linear Regression with Symmetric Triangular Fuzzy Numbers

Xilong Liu and Yizeng Chen

School of Management, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Yizeng Chen; mfcyz@shu.edu.cn

Received 17 July 2013; Revised 12 October 2013; Accepted 14 October 2013

Academic Editor: T. Warren Liao

A systematic approach is proposed to optimize h value for fuzzy linear regression (FLR) analysis using minimum fuzziness criteria with symmetric triangular fuzzy numbers (TFNs). Firstly, a new concept of credibility is defined to evaluate the performance of FLR models with different h values when a set of sample data pairs is given. Secondly, based on the defined concept of credibility, a programming model is formulated to optimize the value of h. Finally, both the numerical study and the real application show that the approach proposed in this paper is effective and efficient; that is, optimal value for h can be determined definitely with respect to a set of given sample data pairs.

1. Introduction

Statistic regression analysis and fuzzy regression analysis are two types of methods underlying different philosophies to assess the functional relationship between the dependent and independent variables and determine the best-fit model for describing the relationship, by exploiting the knowledge from the given input-output data pairs. In statistical regression analysis, deviations between the observed values and the estimates are assumed to be random errors disturbed by a probabilistic distribution. Different from statistic regression analysis, in fuzzy regression analysis, the deviations are attributed to the imprecision of the observed values and/or the indefiniteness of model structure. Tanaka et al. [1] firstly proposed fuzzy linear regression (FLR) analysis using the fuzzy functions defined by Zadeh's extension principle [2], in which the observed values can differ from the estimated values to a certain degree of belief [3]. Thus, the uncertainty in this type of regression model becomes fuzziness, not randomness, and the disturbance is incorporated into the fuzzy coefficients, and the final objective is to adjust the fuzzy coefficients from the available sample data pairs.

According to [3], the existing FLR methods can be roughly classified into the following two categories based on criterion function, that is, FLR methods using minimum fuzziness criteria and FLR methods using fuzzy least-squares criteria. By using the first category of FLR methods, FLR model can be built by minimizing the system vagueness. The first FLR method in [1] was extended by using other types of fuzzy coefficients, including general LP-type fuzzy coefficients [4], exponential fuzzy coefficients [5], and triangular fuzzy coefficients [6, 7], Chen et al. depended on symmetric triangular fuzzy numbers to study determination Method for Parameters of Rock's shear strength through least absolute linear regression, and the analysis of practical engineering computation, and comparison to other methods shows that the method is reasonable [8], trapezoidal fuzzy coefficients [9], and Kheirfam and Verdegay extend the dual simplex method to a type of fuzzy linear programming problem involving symmetric trapezoidal fuzzy numbers. The results obtained lead to a solution for fuzzy linear programming problems and the optimal value function with fuzzy coefficients [10]. And interval regression, where a model with interval coefficients is assumed, is regarded as the simplest version of fuzzy regression analysis [11–13]. Some fuzzy nonlinear regression approaches also were proposed [3, 7]; a research indicated that prediction performance of the nonlinear multiple regression model is higher than that of the fuzzy inference system model [14]. Based on the other direction of FLR methods, FLR model will be built

by minimizing fuzzy distance between the predicted outputs and the observed outputs [15–17]. Celmiņš [15] has dealt with quadratic membership functions based on least squares fitting with indicators of discord, data spread dilator, and so forth, and Diamond [16] proposed models for least squares fitting for crisp input fuzzy output and for fuzzy input-output where the distance of fuzzy numbers is defined to measure the best fit for models. Chang [17] formulated fuzzy least-squares regression model by defining the weighted fuzzy arithmetic (WFA). Lv et al. proposed a novel least squares support vector machine- (LSSVM-) based ensemble learning paradigm to predict NO_x emission of a coal-fired boiler using real operation data. The result shows that the new soft FCM-LSSVM-PLS ensemble method can predict NO_x emission accurately [18].

Regarding the first category of FLR methods, the value of h determines the range of the possibility distribution of the fuzzy parameters [1, 3–8, 19], so it is important to select a suitable value for h in FLR analysis. Moskowitz and Kim [20] studied the relationship among the h value, membership function shape, and the spreads of fuzzy parameters in FLR with symmetric fuzzy numbers, they also developed a general approach to assess the proper h parameter values. Their studies showed that the system fuzziness will increase with the augment of h value. And Tanaka and Watada [19] suggested that the selection of the h value should be based on the sufficiency of the collected dataset. When the dataset is sufficiently large, $h = 0$ should be used and is increased along with the decreasing volume of the collected data. However, both did not suggest how to get an optimal h value for the FLR model when a sample dataset is given. In practical situations, the value of h is usually subjectively preselected by the decision makers (DMs) [17, 18].

In fact, if a larger value is given to h, the FLR using triangular fuzzy coefficients tends to yield large unnecessary fuzziness and estimated parameters with too large aspiration, which leads to the fuzzy predictive interval too fuzzier and has no operational definition or interpretation [8]. On the other hand, if a smaller h value is used, the FLR tends to yield very lower membership degree, which leads to the very narrow fuzzy predictive interval and the reliability of the FLR model is doubtable. Therefore, it is necessary to develop a systematic approach to help DMs determine the optimal h value for the FLR using minimum fuzziness criteria. To tackle the problem, with the suggestion from Tanaka and Watada [19], a new concept of credibility is introduced to measure the performance of the FLR models with different h values in this paper, based on which a systematic approach is formulated to optimize h values for FLR using minimum fuzziness.

The rest of the paper is organized as follows. In Section 2, Tanaka's FLR method is described. In Section 3, the concept of credibility is proposed to measure the performance of FLR models with different h values. In Section 4, a systematic approach is formulated to optimize the h value for FLR with symmetric triangular fuzzy numbers (TFNs). In Section 5, a numerical example is used to show how the optimal value can be determined using the approach proposed in this paper. In Section 6, a real application is conducted to illustrate the effectiveness of the approach proposed in this paper. The conclusions are drawn in Section 7.

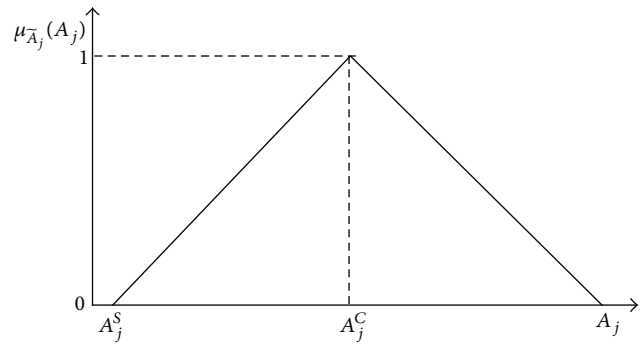


Figure 1: Symmetric triangular fuzzy number.

2. FLR with Symmetric TFNs

As in a FLR analysis, the explained variable is assumed to be a linear combination of the explanatory variables. This relationship should be obtained from a sample of m observations $\{(y_1, \mathbf{x}_1), \ldots, (y_i, \mathbf{x}_i), \ldots, (y_m, \mathbf{x}_m)\}$, where y_i is the ith crisp observation and $\mathbf{x}_i = \{x_{i0}, x_{i1}, \ldots, x_{ij}, \ldots, x_{in}\}$ is the ith crisp input vector. Moreover, $x_{i0} = 1 \forall i$, and x_{ij} is the observed value for the jth variable in the ith case of the sample. In particular, the fuzzy linear function has to be estimated as follows:

$$\widetilde{y}_i = f(\mathbf{x}_i) = \widetilde{A}_0 + \widetilde{A}_1 x_{i1} \cdots \widetilde{A}_j x_{ij} \cdots + \widetilde{A}_n x_{in} = \sum_{j=0}^{n} \widetilde{A}_j x_{ij}, \tag{1}$$

where $\widetilde{y}_i$ is the fuzzy estimation of y_i. And $\widetilde{A}_j$, $j = 0, 1, \ldots, n$, are fuzzy coefficients in the terms of symmetric TFNs and can be uniquely defined by $\widetilde{A}_j = (a_j^S, a_j^C)$, $j = 0, 1, \ldots, n$. Here, a_j^S is the spread value, and a_j^C is the centre value of $\widetilde{A}_j$ (see Figure 1).

The goal in the fuzzy linear regression is to determine $f(\mathbf{x}_i)$ by minimizing the system vagueness subject to the following inclusion condition [1, 19]:

$$y_i \in [f(\mathbf{x}_i)]^h, \quad i = 1, 2, \ldots, m, \tag{2}$$

where $f(\mathbf{x}_i)^h$ is the h-level set of the fuzzy output from the linear fuzzy model $f(\mathbf{x}_i)$ corresponding to the input vector $\mathbf{x}_i$. Since h-level set of fuzzy numbers are intervals, (2) can be further given as follows by using Interval arithmetic:

$$\sum_{j=0}^{n} a_j^C x_{ij} - (1-h) \sum_{j=0}^{n} a_j^S \left| x_{ij} \right| \leq y_{ij} \leq \sum_{j=0}^{n} a_j^C x_{ij} + (1-h) \sum_{j=0}^{n} a_j^S \left| x_{ij} \right|, \quad i = 1, 2, \ldots, m. \tag{3}$$

The system fuzziness in (1), Δ, can be given as

$$\Delta = \sum_{i=1}^{m} \Delta \widetilde{y}_i \tag{4}$$

in which $\Delta \widetilde{y}_i$ is the fuzziness associated with $\widetilde{y}_i$ and can be given as

$$\Delta \widetilde{y}_i = \sum_{j=0}^{n} a_j^S \left| x_{ij} \right|. \tag{5}$$

Henceforth, according to [1, 19], $\widetilde{A}_j$, $j = 0, 1, \ldots, n$, in the form of symmetric TFNs, can be determined by solving the following optimal programming model:

$$\min \ \Delta = \sum_{i=1}^{m} \sum_{j=0}^{n} a_j^S \left| x_{ij} \right| \tag{6a}$$

subject to

$$(1-h) \sum_{j=0}^{n} a_j^S \left| x_{ij} \right| - \sum_{j=0}^{n} a_j^C x_{ij} \geq -y_{ij}, \quad i = 1, \ldots, m, \tag{6b}$$

$$(1-h) \sum_{j=0}^{n} a_j^S \left| x_{ij} \right| + \sum_{j=0}^{n} a_j^C x_{ij} \geq y_{ij}, \quad i = 1, \ldots, m, \tag{6c}$$

$$a_j^S \geq 0, \quad j = 0, 1, \ldots, n \tag{6d}$$

in which the constraints (6b) and (6c) are corresponding to inclusion condition in (2), and the constraint (6d) guarantees that the spread values of $\widetilde{A}_j$, $j = 0, 1, \ldots, n$, are nonnegative.

3. Credibility Measure for the FLR Model

As we can see from (6a) to (6d), the value of h determines the range of the possibility distribution of the fuzzy parameters, so it is important to select a suitable h value for FLR. To do this, the first problem to be solved is how to evaluate the FLR models with different h values in (1). Therefore, a new concept of credibility measurement is introduced in this section, based on which the FLR models associated with different h values can be assessed.

Assume that h_1 and h_2 are any two values for h and $0 \leq h_1 < 1$ and $0 \leq h_2 < 1$. And we denote that $\widetilde{A}_j^1$ and $\widetilde{A}_j^2$, $j = 0, 1, \ldots, n$, are two sets of symmetric TFNs with respect to h_1 and h_2, respectively, and $\widetilde{y}_i^1$ and $\widetilde{y}_i^2$ are the corresponding estimations of y_i from (1). From (1), $\widetilde{y}_i^1$ and $\widetilde{y}_i^2$ are calculated as

$$\widetilde{y}_i^1 = \sum_{j=0}^{n} \widetilde{A}_j^1 x_{ij}, \tag{7}$$

$$\widetilde{y}_i^2 = \sum_{j=0}^{n} \widetilde{A}_j^2 x_{ij}. \tag{8}$$

Now, we are interested in which one, out of $\widetilde{y}_i^1$ and $\widetilde{y}_i^2$, is better as the estimation of y_i. That is to say, which one, out of h_1 and h_2, is better to be used to build a FLR model when the sample set of data pairs is given? To deal with the problem, in our opinion, two factors, that is, the estimating fuzziness $\Delta \widetilde{y}_i$ and the membership degree $\mu_{\widetilde{y}_i}(y_i)$, should be taken into account. In general, the smaller $\Delta \widetilde{y}_i$ is and the higher $\mu_{\widetilde{y}_i}(y_i)$ is, the better the performance of $\widetilde{y}_i$ in representing y_i will be. To illustrate our point of view, two special cases are considered firstly as follows.

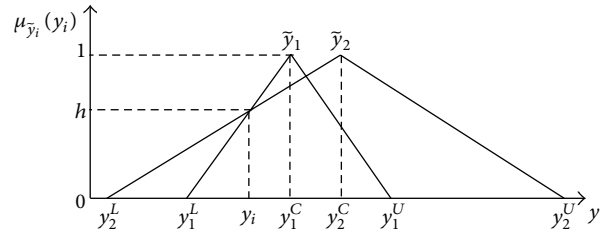


Figure 2: $\mu_{\widetilde{y}_i^1}(y_i) = \mu_{\widetilde{y}_i^2}(y_i)$.

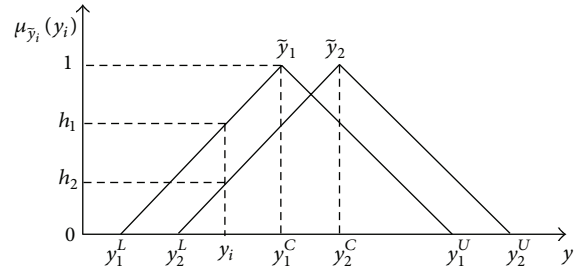


Figure 3: $\Delta \widetilde{y}_i^1 = \Delta \widetilde{y}_i^2$.

(1) If $\mu_{\widetilde{y}_i^1}(y_i)$ is equal to $\mu_{\widetilde{y}_i^2}(y_i)$, it is clear that, out of $\widetilde{y}_i^1$ and $\widetilde{y}_i^2$, the performance of one with smaller fuzziness is better than that of the other with larger fuzziness in representing y_i. As shown in Figure 2, $\mu_{\widetilde{y}_i^1}(y_i) = \mu_{\widetilde{y}_i^2}(y_i)$ and $\Delta \widetilde{y}_i^1 < \Delta \widetilde{y}_i^2$; therefore, $\widetilde{y}_i^1$ is better than $\widetilde{y}_i^2$ as the estimation of y_i.

(2) If $\Delta \widetilde{y}_i^1$ is equal to $\Delta \widetilde{y}_i^2$, it is doubtless that, out of $\widetilde{y}_i^1$ and $\widetilde{y}_i^2$, the performance of one with higher membership degree is better than that of the other with lower membership degree in representing y_i. As shown in Figure 3, $\Delta \widetilde{y}_i^1 = \Delta \widetilde{y}_i^2$ and $\mu_{\widetilde{y}_i^1}(y_i) > \mu_{\widetilde{y}_i^2}(y_i)$; therefore, $\widetilde{y}_i^1$ is better than $\widetilde{y}_i^2$ as the estimation of y_i.

Now let us consider a more general situation that neither $\mu_{\widetilde{y}_i^1}(y_i)$ is equal to $\mu_{\widetilde{y}_i^2}(y_i)$ nor $\Delta \widetilde{y}_i^1$ is equal to $\Delta \widetilde{y}_i^2$, as shown in Figure 4. To deal with this problem, a new concept of credibility measure is introduced in this paper. We denote the credibility of $\widetilde{y}_i$ in representing y_i as z_i, which is expressed as

$$z_i = \frac{\mu_{\widetilde{y}_i}(y_i)}{\Delta \widetilde{y}_i}. \tag{9}$$

The higher z_i is, the better the performance of $\widetilde{y}_i$ in representing y_i will be. Obviously, the scenarios of (1) and (2) are two special cases of (9) respectively.

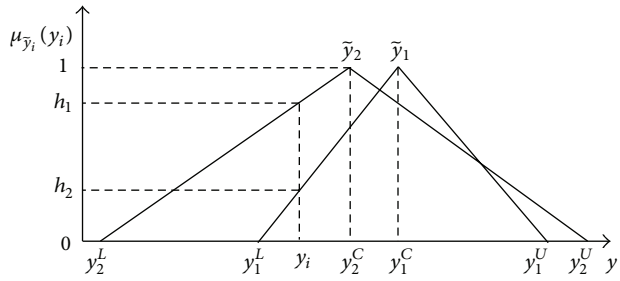


Figure 4: $\Delta\widetilde{y}_i^1 \neq \Delta\widetilde{y}_i^2$ and $\mu_{\widetilde{y}_i^1}(y_i) \neq \mu_{\widetilde{y}_i^2}(y_i)$.

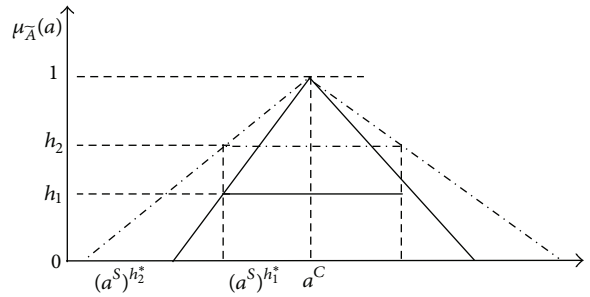


Figure 5: The case when $h_2 \geq h_1$.

As a result, based on (9), the total credibility of all sample data, z, can be obtained to assess the performance of the obtained FLR model in (1), which can be calculated as follows:

$$z = \sum_{i=1}^{m} z_i = \sum_{i=1}^{m} \frac{\mu_{\widetilde{y}_i}(y_i)}{\Delta\widetilde{y}_i}. \tag{10}$$

The higher z is, the better of the performance of the FLR model will be. This will help us select the optimal value of h for a FLR model.

4. Formulating a Systematic Approach to Optimizing h Value

Based on the concept of credibility measure for FLR models introduced in Section 3, a systematic approach is proposed to optimize h value for FLR models in this Section.

As shown in Figure 5, we denote the optimal solution of (6a), (6b), (6c), and (6d) with respect to h_1 as $\widetilde{A}_j^{h_1^*} = ((a_j^S)^{h_1^*}, a_j^C)$, $j = 0, 1, \ldots, n$, and $\Delta^{h_1^*}$, and the corresponding fuzziness as $\Delta\widetilde{y}_i^{h_1^*}$, $i = 1, \ldots, m$. According to Moskowitz and Kim [20], the optimal solution of (6a), (6b), (6c), and (6d) and the corresponding fuzziness with regard to h_2 can be obtained as

$$\begin{aligned} \widetilde{A}_j^{h_2^*} &= \left(\frac{1-h_1}{1-h_2}\left(a_j^S\right)^{h_1^*}, a_j^C\right), \quad j = 0, 1, \ldots, n, \\ \Delta^{h_2^*} &= \frac{1-h_1}{1-h_2}\Delta^{h_1^*}, \\ \Delta\widetilde{y}_i^{h_2^*} &= \frac{1-h_1}{1-h_2}\Delta\widetilde{y}_i^{h_1^*}, \quad i = 1, \ldots, m. \end{aligned} \tag{11}$$

Table 1: Testing dataset.

y	x
14	2
16	4
14	6
18	8
18	10
22	12
18	16
22	18

This indicates that the central value a_{ij}^C of each $\widetilde{A}_{ij}$ will keep constant when the h value changes from h_1 to h_2, while the spread value a_{ij}^S of each $\widetilde{A}_{ij}$ and the objective function Δ and the fuzziness $\Delta\widetilde{y}_i$ $(i = 1, \ldots, m)$ become $(1-h_1)/(1-h_2)$ times simultaneously.

With respect to h_1, $(\mu_{\widetilde{y}_i}(y_i))^{h_1}$, $i = 1, \ldots, m$, are calculated as follows:

$$\left(\mu_{\widetilde{y}_i}(y_i)\right)^{h_1} = 1 - \frac{\left|y_i^C - y_i\right|}{\Delta\widetilde{y}_i^{h_1}} = 1 - q_i^{h_1}, \quad i = 1, \ldots, m \tag{12}$$

in which $q_i^{h_1}$ is given as

$$q_i^{h_1} = \frac{\left|y_i^C - y_i\right|}{\Delta\widetilde{y}_i^{h_1}} = \frac{\left|\sum_{j=0}^{n}\left(a_j^C\right)x_{ij} - y_i\right|}{\sum_{j=0}^{n}\left(a_j^S\right)^{h_1}\left|x_{ij}\right|}, \quad i = 1, \ldots, m. \tag{13}$$

Therefore, according to (8), the estimating credibility for all sample data with regard to h_1 can be expressed as

$$z_i^{h_1} = \frac{\left(\mu_{\widetilde{y}_i}(y_i)\right)^{h_1}}{\Delta\widetilde{y}_i^{h_1}} = \frac{1 - q_i^{h_1}}{\Delta\widetilde{y}_i^{h_1}}, \quad i = 1, \ldots, m. \tag{14}$$

Therefore, from (9), the total credibility for the FLR model with respect to h_1 can be obtained as

$$z^{h_1} = \sum_{i=1}^{m} z_i^{h_1} = \sum_{i=1}^{m} \frac{1 - q_i^{h_1}}{\Delta\widetilde{y}_i^{h_1}}. \tag{15}$$

From (12), with regard to h_2, we have

$$\begin{aligned} q_i^{h_2} &= \frac{\left|y_i^C - y_i\right|}{\Delta\widetilde{y}_i^{h_2}} = \frac{\left|y_i^C - y_i\right|}{\left(\left(1-h_1\right)/\left(1-h_2\right)\right)\Delta\widetilde{y}_i^{h_1}} \\ &= \frac{1-h_2}{1-h_1}q_i^{h_1}, \quad i = 1, \ldots, m. \end{aligned} \tag{16}$$

Table 2: Corresponding results with symmetric triangular fuzzy coefficients ($h = 0$).

i	1	2	3	4	5	6	7	8
$\Delta\widetilde{y}_i^0$	1.2500	1.5000	1.7500	2.0000	2.2500	2.5000	2.7500	3.0000
q_i^0	0.6000	1.0000	1.0000	0.5000	0.1111	1.0000	1.0000	0.0000
p_i^0	0.4800	0.6667	0.5714	0.2500	0.0494	0.4000	0.3636	0.0000
$\left(\mu_{\widetilde{y}_i}(y_i)\right)^0$	0.4000	0.0000	0.0000	0.5000	0.8889	0.0000	0.0000	1.0000
z_i^0	0.3200	0.0000	0.0000	0.2500	0.3951	0.0000	0.0000	0.3333

Therefore, according to (11), $(\mu_{\widetilde{y}_i}(y_i))^{h_2}$ with regard to h_2 can be calculated as

$$\begin{aligned}\left(\mu_{\widetilde{y}_i}(y_i)\right)^{h_2} &= 1 - q_i^{h_2} = 1 - \frac{1-h_2}{1-h_1} q_i^{h_1} \\ &= 1 - q_i^{h_1} + \frac{h_2 - h_1}{1-h_1} q_i^{h_1} \\ &= \left(\mu_{\widetilde{y}_i}(y_i)\right)^{h_1} + \frac{h_2 - h_1}{1-h_1} q_i^{h_1}, \quad i = 1, \ldots, m.\end{aligned} \tag{17}$$

Henceforth, according to (8), (10), and (16), the estimating credibility for all sample data pairs with regard to h_2 can be expressed as

$$\begin{aligned}z_i^{h_2} &= \frac{\left(\mu_{\widetilde{y}_i}(y_i)\right)^{h_2}}{\Delta\widetilde{y}_i^{h_2}} \\ &= \frac{\left(\mu_{\widetilde{y}_i}(y_i)\right)^{h_1} + \left((h_2 - h_1)/(1-h_1)\right) q_i^{h_1}}{\left((1-h_1)/(1-h_2)\right) \Delta\widetilde{y}_i^{h_1}} \\ &= \frac{1-h_2}{1-h_1} z_i^{h_1} + \frac{(1-h_2)(h_2-h_1)}{(1-h_1)^2} p_i^{h_1}, \quad i = 1, \ldots, m\end{aligned} \tag{18}$$

in which

$$p_i^{h_1} = \frac{q_i^{h_1}}{\Delta\widetilde{y}_i^{h_1}} = \frac{\left|y_i^C - y_i\right|}{\left(\Delta\widetilde{y}_i^{h_1}\right)^2}, \quad i = 1, \ldots, m. \tag{19}$$

Consequently, the estimating credibility for the FLR model in (1) with regard to h_2 can be obtained as

$$\begin{aligned}z^{h_2} &= \sum_{i=1}^{m} z_i^{h_2} = \sum_{i=1}^{m} \left(\frac{1-h_2}{1-h_1} z_i^{h_1} + \frac{(1-h_2)(h_2-h_1)}{(1-h_1)^2} p_i^{h_1} \right) \\ &= \frac{1-h_2}{1-h_1} z^{h_1} + \frac{(1-h_2)(h_2-h_1)}{(1-h_1)^2} p^{h_1}\end{aligned} \tag{20}$$

in which

$$p^{h_1} = \sum_{i=1}^{m} p_i^{h_1}. \tag{21}$$

For similarity, we denote $h_1 = 0$ and $h_2 = h$, (11), (16), (17), (18), and (20) can be rewritten as

$$\widetilde{A}_j^h = \left(\frac{1}{1-h} \left(a_j^S\right)^0, a_j^C \right), \quad j = 0, 1, \ldots, n,$$

$$\Delta^h = \frac{1}{1-h} \Delta^0, \tag{22}$$

$$\Delta\widetilde{y}_i^h = \frac{1}{1-h} \Delta\widetilde{y}_i^0, \quad i = 1, \ldots, m,$$

$$q_i^h = (1-h)\, q_i^0, \tag{23}$$

$$\left(\mu_{\widetilde{y}_i}(y_i)\right)^h = \left(\mu_{\widetilde{y}_i}(y_i)\right)^0 + h q_i^0, \tag{24}$$

$$z_i^h = -p_i^0 h^2 + \left(p_i^0 - z_i^0\right) h + z_i^0, \tag{25}$$

$$z^h = -p^0 h^2 + \left(p^0 - z^0\right) h + z^0. \tag{26}$$

Based on (26), the optimal h value for the FLR model in (1) can be obtained by maximizing the estimating credibility; that is, the optimal value for h can be obtained by solving the following programming model:

$$\max \quad z^h = -p^0 h^2 + \left(p^0 - z^0\right) h, \tag{27a}$$

$$\text{s.t.} \quad 0 \le h < 1. \tag{27b}$$

It is obviously that the programming model in (27a) to (27b) is quadratic, and many kinds of optimization algorithms, such as the gradient descent method, can be used to solve the previous nonlinear programming problem.

5. Numerical Example

The numerical example in [21] is used to how the optimal h value can be obtained using the approach proposed in this paper. The eight testing data pairs are listed in Table 1.

When h is specified as 0, the fuzzy coefficients in terms of symmetric TFNs can be obtained by solving the programming model in (6a), (6b), (6c), and (6d) as $\widetilde{A}_0^0 = (1.0000, 12.0000)$ and $\widetilde{A}_1^0 = (0.1250, 0.6250)$. And the corresponding FLR model is given as

$$\widetilde{y}^0 = (1.0000, 12.0000) + (0.1250, 0.6250)\, x. \tag{28}$$

Accordingly, the fuzziness $\Delta\widetilde{y}_i^0$, parameters q_i^0 and p_i^0, membership degree $(\mu_{\widetilde{y}_i}(y_i))^0$, and credibility z_i^0, $i = 1, \ldots, m$, are calculated, respectively, as shown in Table 2.

Table 3: Testing data set.

y	x_1	x_2
133.60	29.50	79.99
137.63	31.30	75.63
147.86	37.60	69.25
196.76	39.90	62.75
220.53	39.90	64.66
223.25	40.30	63.09
233.19	41.50	61.51
265.67	43.60	60.07
335.16	45.70	58.22
411.29	47.80	58.43
460.68	49.50	60.57
477.96	50.10	58.23
474.02	50.20	58.03
466.80	49.90	57.53
466.16	50.00	55.68
469.80	50.00	55.24
468.95	50.00	54.51
476.24	50.90	50.08
499.39	53.10	50.05
521.20	55.20	49.72

From Table 2, z^0 is calculated as 1.2984 and p^0 as 2.7811. According to (26), the total credibility of the FLR model with respect to h can be expressed as follows:

$$z^h = -2.7811h^2 + 1.4827h + 1.2984. \tag{29}$$

According to (27a) and (27b), the optimal value h^* is given as 0.2666, and the optimal coefficients in the form of symmetric case are obtained as $\widetilde{A}_0^* = (1.3635, 12.0000)$ and $\widetilde{A}_1^* = (0.1704, 0.6250)$. Therefore, the optimal FLR model is given as

$$\widetilde{y}^* = (1.3635, 12.0000) + (0.1704, 0.6250)\, x. \tag{30}$$

The total credibility of the model in (30) z^* is calculated as 1.4960, which is higher than that of the model in (28), that is, 1.2984.

The changes of the fuzziness $\Delta\widetilde{y}_i^h$, the membership degree $(\mu_{\widetilde{y}_i}(y_i))^h$, the credibility z_i^h, $i = 1, \ldots, m$, and the total credibility z^h with different h values are depicted in Figures 6, 7, 8, and 9, respectively. Notably, the vertical axis is logarithmic scale in Figure 6 to enable us to see small changes in fuzziness among eight sample data, and the membership degrees for sample data 2, 3, 6, and 7 are overlapped in Figure 7.

From Figure 6, we can see that the fuzziness of all sample data become larger with the augment of h value; especially when h value is close to 1, the fuzziness will be extremely larger. From Figure 7, it can be found that the membership degree of all sample data becomes higher with the increase of h value, and the membership degree will be close to 1 when h value is close to 1. It is clearly shown in Figure 8 that when

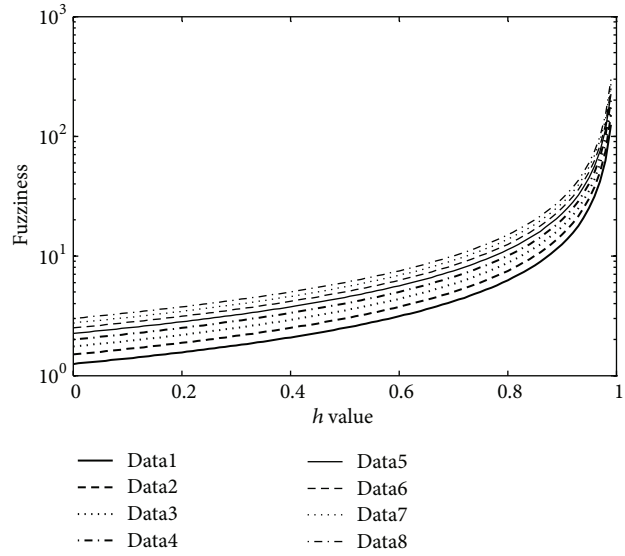


Figure 6: Fuzziness of the eight sample data.

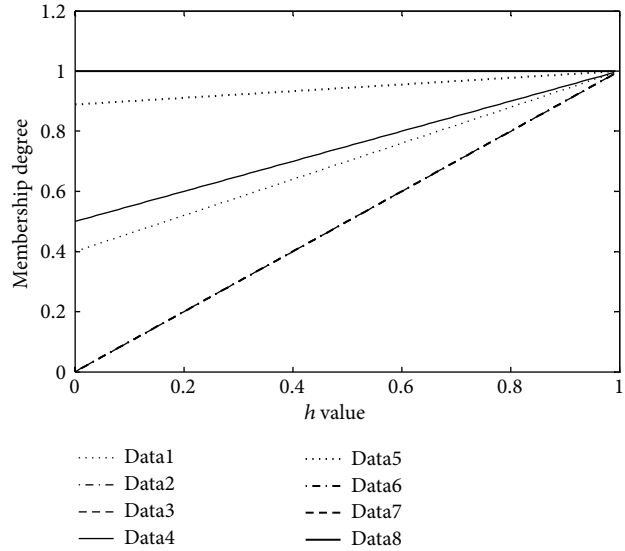


Figure 7: Membership degree of the eight sample data.

h value is close to 1, the credibility measures for all sample data will be close to zero due to the extremely larger of the fuzziness (see Figure 7). Figure 9 shows that when h value is 0.2666, the total credibility of the FLR model will achieve the maximum, based on which the best FLR model can be obtained.

To further demonstrate the effectiveness of the approach proposed in this paper, another numerical example with two independent variables is given as follows. The twenty testing datasets are listed in Table 3.

According to (27a) and (27b), the optimal value h^* is given as 0.3184. When h is specified as 0, 0.3184, and 0.5, respectively, the fuzzy coefficients in terms of symmetric TFNs and the corresponding total credibility can be obtained

Table 4: The corresponding results.

h value	Fuzzy linear model	Total credibility
0.3148	$\tilde{y}^{*} = (-610.8251, 0.0000) + (19.1484, 0.0000)\, x_1 + (1.4018, 1.2301)\, x_2$	0.1371
0.0	$\tilde{y}^{0} = (-610.8233, 0.0000) + (19.1484, 0.0000)\, x_1 + (1.4017, 0.8429)\, x_2$	0.1082
0.5	$\tilde{y}^{0.5} = (-610.8240, 0.0000) + (19.1484, 0.0000)\, x_1 + (1.4017, 1.6857)\, x_2$	0.1271

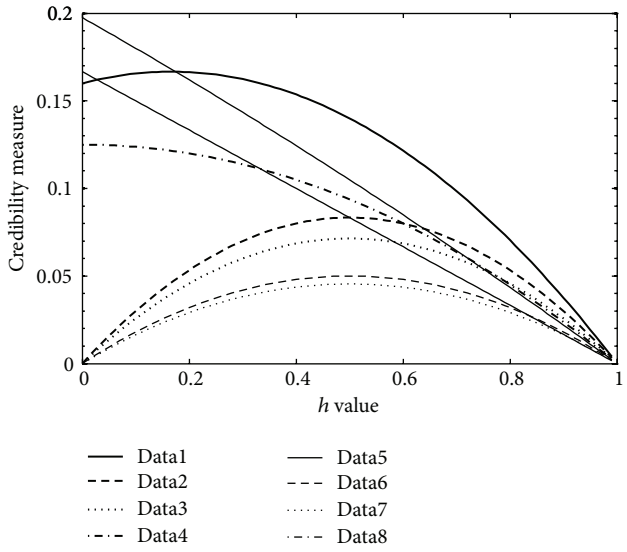


Figure 8: Credibility of the eight sample data.

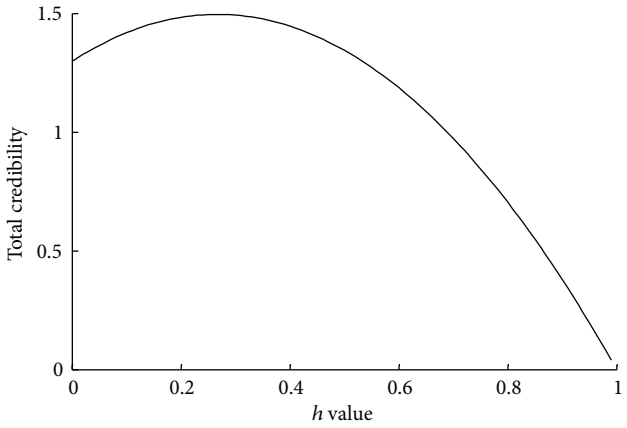


Figure 9: Total credibility of the FLR models.

by solving the programming model in (6a) , (6b), (6c), and (6d). The results are summarized in Table 4.

From Table 4, we can see that when h value is set to 0.3148, the total credibility of the FLR model will achieve the maximum, that is, 0.1371, which indicates that the approach proposed in this paper is effective and reasonable.

Table 5: The experiment results.

Run no.	x (g/cm^3)	y (horsepower)
1	0.760	5.1
2	0.780	6.2
3	0.820	7.5
4	0.815	7.0
5	0.790	7.1
6	0.785	6.8
7	0.770	5.9
8	0.765	5.6
9	0.788	9.0
10	0.769	5.9

Table 6: The FLR models with the corresponding total credibility.

h value	Fuzzy linear model	Total credibility
0.2057	$y = (-25.6000, 0.8765) + (40.6962, 0.0000)x$	7.2785
0	$y = (-25.6000, 0.6962) + (40.6962, 0.0000)x$	6.7903
0.5	$y = (-25.6000, 1.3924) + (40.6962, 0.0000)x$	6.2793

6. Real Application

To investigate the effectiveness of the approach proposed in this paper, modeling welding process for electronic manufacturing using fuzzy linear regression was studied. An electronic company is a famous OEM company of printed circuit board (PCB) in PR China. The engineers in this company wanted to improve the welding quality by investigating the relationship between the pull strength (y) of welding line and the proportion of colophony (x) in welding fluid. And an engineering experiment was conducted and the experiment results are shown in Table 5.

With the use of the 10 experimental datasets pairs, the optimal h value is calculated as 0.2057. If h value is given as 0.2057 (the optimal value), 0, and 0.5, respectively, the FLR models with the corresponding total credibility are summarized in Table 6.

It is not surprise that the total credibility of the FLR model with the optimal h value achieves the highest among the three FLR models (see Table 6). In fact, when the optimal h value is used, the total credibility of the FLR model will improve 7.19% comparing with $h = 0$ and 15.91% comparing with $h = 0.5$.

To further investigate the modeling performance of the approach proposed in this paper, we divided the sample data pairs to two groups: one group is for modeling; that is, 80% of all the sample data pairs were used to build the FLR model; the other group is for testing; that is, the left 20% of all the

Table 7: Testing results.

Testing data	h^*	$h = h^*$ Total credibility	$h = 0$ Total credibility	$h = 0.5$ Total credibility
9, 10	0.2206	1.5721	1.5434	1.3085
1, 4	0.2820	1.9803	1.5332	1.1472
5, 6	0.2018	0.2841	0.0000	0.2874
2, 9	0.2602	1.8206	2.0378	1.4255
4, 7	0.2761	1.0588	1.2084	1.0614
1, 2	0.3066	2.1009	1.7199	1.6128
2, 5	0.2963	1.4407	1.5948	1.3063
3, 7	0.2743	1.8868	1.2155	1.4581
1, 5	0.2647	1.2111	1.2000	1.2000
6, 10	0.2139	1.5273	1.4654	1.2908
Average		1.4833	1.3518	1.2098

sample data were used to test the performance of the FLR model. Therefore, each time, two datasets were randomly selected from ten datasets as testing datasets while the rest eight datasets were used to develop FLR models when h value is specified as the optimal value, 0 and 0.5, respectively. Their corresponding total credibility was calculated. The previous procedures were repeated for ten times. Table 7 summarizes the testing results.

From Table 7, it can be seen that the predictive performance of the FLR model with the optimal h value is the best among those of the other models. In fact, in general, when the optimal h value is used, the total credibility of the FLR model will improve 9.73% comparing with $h = 0$ and 22.61% comparing with $h = 0.5$. Therefore, the approach proposed in this paper is effective.

7. Conclusions

In this paper, a systematic approach is proposed to select optimal value of h for FLR analysis with symmetric TFNs. Firstly, a new concept of credibility is introduced by the consideration of system fuzziness and membership degree, which can be used to assess the performance of FLR analysis with different h values when a set of sample data pairs is given. Secondly, a procedure to obtain optimal h value is formulated for FLR with symmetric TFN by maximizing the total credibility of FLR models. By using the approach proposed in this paper, the optimal value of h can be determined definitely with respect to a set of sample data pairs. Both the numerical example and real application demonstrate that the approach proposed in this paper is effective and efficient. The further work in this direction involves extending the approach proposed in this paper for FLR analysis with fuzzy outputs, that is, developing a procedure to select optimal h value for FLR with fuzzy observations and applying the proposed approach to practical problems.

Acknowledgments

The work described in this paper is supported by a Grant from the National Nature Science Foundation of Chinese (Project no. NSFC 71272177/G020902) and the funds of Innovation Program of Shanghai Municipal Education Commission (Project no. 12ZS101).

References

[1] H. Tanaka, S. Uejima, and K. Asai, "Linear regression analysis with fuzzy model," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 12, no. 6, pp. 903–907, 1982.

[2] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978.

[3] D. Zhang, L.-F. Deng, K.-Y. Cai, and A. So, "Fuzzy nonlinear regression with fuzzified radial basis function network," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 742–760, 2005.

[4] A. Bárdossy, "Note on fuzzy regression," *Fuzzy Sets and Systems*, vol. 37, no. 1, pp. 65–75, 1990.

[5] H. Tanaka, H. Ishibuchi, and S. Yoshikawa, "Exponential possibility regression analysis," *Fuzzy Sets and Systems*, vol. 69, no. 3, pp. 305–318, 1995.

[6] K. K. Yen, S. Ghoshray, and G. Roig, "A linear regression model using triangular fuzzy number coefficients," *Fuzzy Sets and Systems*, vol. 106, no. 2, pp. 167–177, 1999.

[7] Y. Chen, J. Tang, R. Y. K. Fung, and Z. Ren, "Fuzzy regression-based mathematical programming model for quality function deployment," *International Journal of Production Research*, vol. 42, no. 5, pp. 1009–1027, 2004.

[8] Z. B. Chen, B. Lin, and Q. Fang, "Study on determination method for parameters of Rock's shear strength through least absolute linear regression based on symmetric triangular fuzzy numbers," *Applied Mechanics and Materials*, vol. 170, pp. 804–807, 2012.

[9] R. Y. K. Fung, Y. Chen, and J. Tang, "Estimating the functional relationships for quality function deployment under uncertainties," *Fuzzy Sets and Systems*, vol. 157, no. 1, pp. 98–120, 2006.

[10] B. Kheirfam and J.-L. Verdegay, "The dual simplex method and sensitivity analysis for fuzzy linear programming with symmetric trapezoidal numbers," *Fuzzy Optimization and Decision Making*, vol. 12, no. 2, pp. 171–189, 2013.

[11] P. Diamond and H. Tanaka, "Fuzzy regression analysis," in *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, vol. 1 of *Handbooks of Fuzzy Sets Series*, pp. 349–387, Kluwer Academic, Boston, Mass, USA, 1998.

[12] H. Tanaka and H. Lee, "Interval regression analysis by quadratic programming approach," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 4, pp. 473–481, 1998.

[13] C. Hwang and D. H. Hong, "Analyzing the nonlinear time series of turbulent flows with kernel interval regression machine," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 13, no. 4, pp. 365–380, 2005.

[14] S. Yagiz and C. Gokceoglu, "Application of fuzzy inference system and nonlinear regression models for predicting rock brittleness," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2265–2272, 2010.

[15] A. Celmiņš, "Least squares model fitting to fuzzy vector data," *Fuzzy Sets and Systems*, vol. 22, no. 3, pp. 245–269, 1987.

[16] P. Diamond, "Fuzzy least squares," *Information Sciences*, vol. 46, no. 3, pp. 141–157, 1988.

[17] Y.-H. O. Chang, "Hybrid fuzzy least-squares regression analysis and its reliability measures," *Fuzzy Sets and Systems*, vol. 119, no. 2, pp. 225–246, 2001.

[18] Y. Lv, J. Liu, T. Yang, and D. Zeng, "A novel least squares support vector machine ensemble model for emission prediction of a coal-fired boiler," *Energy*, vol. 55, pp. 319–329, 2013.

[19] H. Tanaka and J. Watada, "Possibilistic linear systems and their application to the linear regression model," *Fuzzy Sets and Systems*, vol. 27, no. 3, pp. 275–289, 1988.

[20] H. Moskowitz and K. Kim, "On assessing the H value in fuzzy linear regression," *Fuzzy Sets and Systems*, vol. 58, no. 3, pp. 303–327, 1993.

[21] Y.-H. O. Chang and B. M. Ayyub, "Fuzzy regression methods—a comparative assessment," *Fuzzy Sets and Systems*, vol. 119, no. 2, pp. 187–203, 2001.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 406047, 11 pages
http://dx.doi.org/10.1155/2013/406047

Research Article

Combinatorial Clustering Algorithm of Quantum-Behaved Particle Swarm Optimization and Cloud Model

Mi-Yuan Shan,[1] Ren-Long Zhang,[1] and Li-Hong Zhang[2]

[1] *College of Business Administration, Hunan University, No. 11 Lushan South Road, Changsha 410082, China*
[2] *Liverpool Business School, Liverpool John Moores University, Redmonts Building, Brownlow Hill, Liverpool L3 5UX, UK*

Correspondence should be addressed to Ren-Long Zhang; zhangrenlonggj@163.com

Received 14 April 2013; Revised 22 September 2013; Accepted 7 October 2013

Academic Editor: T. Warren Liao

We propose a combinatorial clustering algorithm of cloud model and quantum-behaved particle swarm optimization (COCQPSO) to solve the stochastic problem. The algorithm employs a novel probability model as well as a permutation-based local search method. We are setting the parameters of COCQPSO based on the design of experiment. In the comprehensive computational study, we scrutinize the performance of COCQPSO on a set of widely used benchmark instances. By benchmarking combinatorial clustering algorithm with state-of-the-art algorithms, we can show that its performance compares very favorably. The fuzzy combinatorial optimization algorithm of cloud model and quantum-behaved particle swarm optimization (FCOCQPSO) in vague sets (IVSs) is more expressive than the other fuzzy sets. Finally, numerical examples show the clustering effectiveness of COCQPSO and FCOCQPSO clustering algorithms which are extremely remarkable.

1. Introduction

Clustering is a popular data analysis method and plays an important role in data mining. Clustering is an important class of unsupervised learning techniques. How to get the best or a satisfying solution quickly has great significance for improving productivity and the development of society. So far, it has been widely applied in many fields, like web mining, pattern recognition, resources allocation, machinelearning, spatial database analysis, artificial intelligence, and so on. The existing clustering algorithms can be simply classified into the following two categories: hierarchical clustering and partitional clustering [1]. Though the K-means algorithm is widely used to solve problems in many areas, KM is very sensitive to initialization, the better centers we choose, the better results we get. Also, it was easily trapped in local optimal [2]. Recently much work was done to overcome these problems. Traditional cluster analysis requires every point of the data set to be assigned into a cluster precisely, so we call it hard clustering. But in fact, most things exist in ambiguity in the attribute, there are no explicit boundaries among the things, and no the nature of either-or. So, the theory of the fuzzy clustering is more suitable for the nature of things, and it can more objectivly reflect the reality. To date, development of clustering has focused on data processing speed and efficiency. How appropriate the number of clusters may affect the clustering results. For example, a smaller number of clusters can help clearly identify the original data structure but key hidden information may not be obtained.

Data clustering was an exploratory or descriptive process for data analysis, where similar objects or data points were identified and grouped into a set of objects called a cluster [3, 4]. The clustering problem was defined as the problem of classifying a collection of objects into a set of natural clusters without any prior knowledge. The main goal of cluster analysis, an unsupervised classification technique was to analyze similarity of data and divide it into several cluster sets of similar data [5]. In terms of internal homogeneity and external separation, data within a cluster had high similarity, but there was less similarity between clusters. As a result, clustering is utilized to divide the data into several cluster sets for analysis and reduce data complexity. Generally, clustering has several steps: (1) proper features were selected from data as the basis for clustering; (2) a suitable clustering algorithm

was selected based on data type; (3) the evaluation principle determines the clustering results, which were provided for experts to interpret [6]. Basically, clustering algorithms can be hierarchical or partitional. Among the unsupervised clustering algorithms, the hierarchical method was most typical [7, 8]. But the problem with partitional methods was necessary to set the number of clusters before clustering as in K-means or K-medoids [9, 10]. In addition, Gath and Geva [11] proposed an unsupervised clustering algorithm based on the combination of fuzzy C-means and fuzzy maximum likelihood estimation. Lorette et al. [12] proposed an algorithm based on fuzzy clustering to dynamically determine the number of clusters in a data set. Furthermore, SYNERACT [13], an alternative approach to ISODATA [14] combining K-means with hierarchical descending approaches, did not require specifying the number of clusters. Based on K-means, other unsupervised clustering algorithms have also been developed, such as X-means and G-means [15, 16]. Omran et al. proposed a new dynamic clustering approach based on binary-PSO (DCPSO) algorithm. Paterlinia and Krink (2006) applied PSO for partitional clustering algorithms partition the data set into a specified number of clusters.These algorithms try to minimize certain criteria and can therefore be treated as optimization problems. In addition, Ouadfel et al. (2010) presented a modified PSO algorithm for automatic image clustering algorithm.Where each cluster groups together similar patterns. According to conducted experiments, the proposed approach outperforms K-means, FCM and KHM algorithm [17, 18]. Such an algorithm can directly generate a suitable number of clusters. In addition, it is critical to have a suitable measure for the effectiveness of a clustering method. Though some measures have been proposed in the area of classification, like Wallace's information measure, few have been proposed for clustering algorithms [19]. The most frequently applied measures were cohesion, within-cluster variance, separation, and between cluster variance [20]. The DBSCAN clustering algorithm can also be used in four applications that were using 2D points, 3D points, 5D points and 2D polygons of real world problems [21]. Those clustering algorithms have been applied in the clustering analysis for many years, but those algorithms are also unilateral.

Now, many improved velocity update rules have been developed, including the inertia weight (W) method [22], constriction factor method [23], guaranteed convergence method [24], improvement social model [25], global-local best value based PSO method, and GLbest-PSO method [26]. In addition, Montes de Oca et al. [27] proposed a novel PSO algorithm combining a number of algorithmic components that showed distinct advantages for optimization speed and reliability in their experimental study.

In addition to these methods, Kennedy and Eberhart [28] proposed the binary-PSO (BPSO), which is applied to optimization problems requiring discrete or binary variables for solutions. A novel optimization algorithm, based on a modified binary-PSO with mutation (MBPSOM) combined with a support vectormachine (SVM), was proposed to select the fault feature variables for fault diagnosis [29]. Chuang et al. [30] presented improved BPSO for feature selection using gene expression data. In addition, Xie et al. [31] suggested the distributive-PSO (DPSO) algorithm. The DPSO algorithm was used to solve problems in which it was easy for the PSO to fall into a local value from which it cannot escape. In particle searching, the DPSO algorithm was utilized to randomly select particles for mutation. Liu et al. [32] proposed PSO with mutation based on similarity, in which similarity and collectivity are introduced. Most recently, Nani and Lumini [33] used the Nearest Neighbor approach for prototype reduction using PSO and they found the Nearest Neighbor approach to be good. In order to achieve a dynamic clustering, where the optimum number of clusters was also determined within the process, the so-called multidimensional PSO (MDPSO) method extended the native structure of PSO particles in such a way that they can make interdimensional pass with a dedicated dimensional PSO process [34]. In addition, Ouadfel et al. [35] presented an automatic clustering using a modified PSO (ACMPSO) algorithm. Also, Martinez et al. [36] proposed a swarm intelligence feature selection algorithm based on the initialization and update of only a subset of particles in the swarm. When compared to many of the other population-based approaches such as other algorithmssuch as GA, ACO, and DE, the convergence rate of the population was much slower for PSO algorithm [37]. Meanwhile,the PSO algorithms was inspired by social behavior among individuals, for instance, bird flocks. Intelligence particles representing a potential problem solution move through the search space.

2. Combinatorial Optimization Algorithm of Cloud Model and Quantum-Behaved Particle Swarm Optimization

2.1. Quantum-Behaved Particle Swarm Optimization. Genetic clustering algorithm is randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. When the clustering algorithm is terminated in the limited time, one is used to find the methodology exhibiting rather poor search precision. Therefore, this unfavorable characteristic may to some extent indicate that the global optimality of the solution given by GA is dubious. Ant Colony clustering algorithm requires a longer period of time to run. At this stage, ants consider only the similarity without considering the density to pick up and down the object data, as results, data objects will be orderly dispersed. Because the ant colony clustering algorithm is a stochastic algorithm. Class of the results of cluster center reselection has the potential to separate from each other class of cluster center. Particle swarm optimization clustering algorithm is a newly rising evolutionary computation technique based on swarm intelligence, pso possesses the better convergent speed and computational precision compared with the traditional algorithms such as GA, PSO, ACO, and DE, it can effectively search out the global optimal solution in the space of solution [38–40]. Let us denote M as the swarm size and define n as the dimensionality of the search space. Particle swarm optimization is a global

optimization algorithm. At the same time, the convergence rate is accelerated by the modified PSO algorithm. From the analysis of quantum physics point of view, the particle state can be described the energy and momentum, measurement results of concrete can be used wave function $\psi(x,t)$. The wave function model of the square is a point particle in the space of probability. Therefore, in the QPSO algorithm, each particle is not expressed by the speed and position but as a quantum state. Particle probability distribution at a location space can be obtained by the probability density function of wave function $|\psi(x,t)|$. Once we determine the probability density function of particle position, we can obtain the probability distribution. The particle position update equation under the guidance of Monte Carlo method is expressed as

$$x_{ij}(t+1) = p_{ij} \pm 0.5 \cdot L_{ij} \cdot \ln\left(\frac{1}{u_{ij}}\right), \tag{1}$$

where u_{ij} is random numbers uniformly distributed at the interval $(0,1)$. The QPSO algorithm employs local version constriction factor method and global version inertia weight method simultaneously to achieve relatively high performance and $p_{ij}(t)$ is local version constriction factor. It is a random point between $p_{\text{best}_i,j}$ and $p_{g\text{best},j}$. The formula of $p_{ij}(t)$ is calculated as

$$\begin{aligned} p_{ij}(t) &= \varphi_{ij} \cdot P_{\text{best}_{i,j}}(t) + \left(1-\varphi_{ij}\right) \cdot G_{\text{best}_i}(t), \\ P_{\text{best}_{i,j}} &= \left[P_{\text{best}_{i,1}}, P_{\text{best}_{i,2}}, \ldots, P_{\text{best}_{i,n}}\right], \\ G_{\text{best}} &= \left[G_{\text{best}_1}, G_{\text{best}_2}, \ldots, G_{\text{best}_n}\right]. \end{aligned} \tag{2}$$

The analysis results are as follows:

$$x_{ij}(t+1) = p_{ij}(t) \pm \beta \cdot \left|p_{ij}(t) - x_{ij}(t)\right| \cdot \ln\left(\frac{1}{u_{ij}}\right). \tag{3}$$

The method introduced the concept of average optimal position M_{best}. M_{best} Variables are expressed in the current moment group at all the best location history as follows:

$$\begin{aligned} P_{\text{best}_i} &= \left[P_{\text{best}_{i,1}}, P_{\text{best}_{i,2}}, \ldots, P_{\text{best}_{i,n}}\right], \\ M_{\text{best}} &= \left[M_{\text{best}_1}, M_{\text{best}_2}, \ldots, M_{\text{best}_n}\right] \\ &= \left[\frac{1}{M}\sum_{i=1}^{M} P_{\text{best}_{i,1}}, \frac{1}{M}\sum_{i=1}^{M} P_{\text{best}_{i,2}}, \ldots, \frac{1}{M}\sum_{i=1}^{M} P_{\text{best}_{i,n}}\right]. \end{aligned} \tag{4}$$

So, the particle position updates formula transformed can be expressed as

$$x_{ij}(t+1) = p_{ij}(t) \pm \beta \cdot \left|M_{\text{best}_j}(t) - x_{ij}(t)\right| \cdot \ln\left(\frac{1}{u_{ij}}\right). \tag{5}$$

Parameter β which is set to linear gradient from 1.0 to 0.5 can generally get good results.

Values k for the uniform distribution random number between $(0,1)$, get basic expressions of QPSO algorithm as follows:

$$\begin{aligned} x_i(t+1) &= \varphi_i \cdot P_{\text{best}_i}(t) + \left(1-\varphi_i\right) \cdot G_{\text{best}}(t) \\ &\quad + \beta \cdot \left|M_{\text{best}}(t) - x_i(t)\right| \cdot \ln\left(\frac{1}{u}\right), \\ &\quad k \geq 0.5, \\ x_i(t+1) &= \varphi_i \cdot P_{\text{best}_i}(t) + \left(1-\varphi_i\right) \cdot G_{\text{best}}(t) \\ &\quad - \beta \cdot \left|M_{\text{best}}(t) - x_i(t)\right| \cdot \ln\left(\frac{1}{u}\right), \\ &\quad k < 0.5. \end{aligned} \tag{6}$$

Necessary and sufficient conditions for a point not algorithm global convergence of the PSO, but it can make the algorithm have better convergence speed and accuracy, the reasons not only involve the analysis of the convergence of the algorithm, but also research on search mechanism and complexity more involved algorithm. How to control search mechanism of the PSO by these design means the controlling of $L_{i,j}(t)$. The first algorithm method is evaluated in detail by the following formula:

$$L_{ij} = 2 \cdot \beta \cdot \left|M_{\text{best}_j} - x_{ij}\right|, \tag{7}$$

or

$$L_{i,j}(t) = 2\beta \cdot \left|p_{i,j}(t) - x_{i,j}(t)\right|. \tag{8}$$

The evolution equation of particle deduction into is expressed as follows:

$$\begin{aligned} X_{i,j}(t+1) &= p_{i,j}(t) \pm \alpha \cdot \left|p_{i,j}(t) - X_{i,j}(t)\right| \\ &\quad \cdot \ln\left[\frac{1}{u_{i,j}(t)}\right], \quad u_{i,j}(t) \sim U(0,1). \end{aligned} \tag{9}$$

The second kind of method is introduced in the algorithm of the average best position as follows:

$$\begin{aligned} &C(t) \cdot C(t) \\ &= \left(C_1(t), C_2(t), \ldots, C_n(t)\right) \\ &= \frac{1}{M}\sum_{i=1}^{M} P_i(t) \\ &= \left(\frac{1}{M}\sum_{i=1}^{M} P_{i,1}(t), \frac{1}{M}\sum_{i=1}^{M} P_{i,2}(t), \ldots, \frac{1}{M}\sum_{i=1}^{M} P_{i,N}(t)\right). \end{aligned} \tag{10}$$

The evolution equation of particle deduction into can be expressed as

$$X_{i,j}(t+1) = p_{i,j}(t) \pm \beta \cdot \left| C_j(t) - X_{i,j}(t) \right| \cdot \ln\left[\frac{1}{u_{i,j}(t)}\right], \quad u_{i,j}(t) \sim U(0,1). \tag{11}$$

The process and the concrete steps of the QPSO algorithm are described as follows:

Step 1. Set $t = 0$, initializing the particle swarm in the position of each particle and individual best position $p_i(0) = x_i(0)$.

Step 2. Calculate the average best position of particle swarm.

Step 3. For each particle group i $(1 \le i \le M)$, Steps 4 to 7.

Step 4. Calculate the fitness of particle in current position $x_i(t)$, according to equation individual best position update particle. If $f[x_i(t)] < f[p_i(t-1)]$, $p_i(t) = x_i(t)$. If not, $p_i(t) = p_i(t-1)$.

Step 5. For the particle, the adaptation values $p_i(t)$ and the global best position values $G(t-1)$ are compared. If $f[p_i(t)] < f[G(t-1)]$, $G(t) = p_i(t)$. If not, $G(t) = G(t-1)$.

Step 6. According to each dimension of the particles, calculate equation at random point position.

Step 7. Calculate the new position calculating particle.

Step 8. If the end condition is not satisfied, $t = t + 1$.

Return to Step 2; otherwise, the algorithm terminates.

2.2. Cloud Model. Cloud model which is proposed based on the traditional fuzzy mathematics and probability statistics theory is a conversion model between qualitative concept and quantitative values. The cloud model which was proposed by Li et al. (1998), is novel uncertainty reasoning technology. Cloud model is an effective tool in uncertain transforming between qualitative concepts and their quantitative expressions. There are various implementation approaches of the cloud model, resulting in different kinds of cloud models. The normal cloud model is the most commonly used model, which is based on the normal distribution and the Gaussian membership function. It can be described as follows. They integrate fuzziness and randomness using digital characteristics, such as expected value E_x, entropy E_n and hyper entropy He.

In the domain of cloud droplets, E_x is the most representative of the qualitative concept, the expectation is that it is the domain of the center value. Entropy E_n is the qualitative concept of randomness and fuzziness, is jointly determined, capable of measuring particle size is represents a qualitative concept. E_n is a measure of the randomness of qualitative concept and reflects the discrete degree of the cloud droplet. The excess entropy E_n is the entropy of the uncertainty of

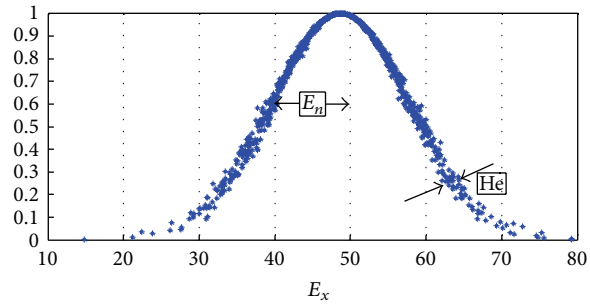


Figure 1: The digital characteristics of the cloud.

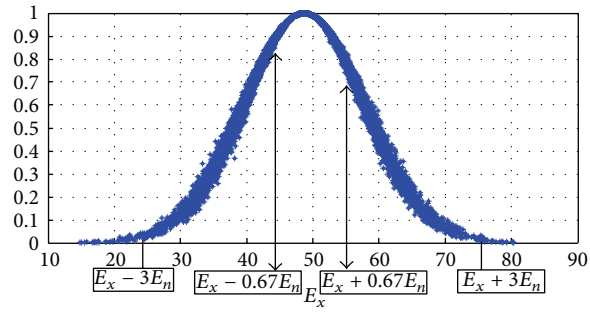


Figure 2: Cloud droplets contribute to qualitative concept.

measurement. The digital characteristics of the cloud are shown in Figure 1.

The hybrid sets which contain fuzziness and probabilities are defined over continuous universe of discourse. Domain $(-\infty, +\infty)$, all elements are in total contribution to the concept of C and expression formula can be expressed as

$$C = \frac{\int_{-\infty}^{+\infty} \mu_A(x)\,dx}{\sqrt{2\pi}E_n} = \frac{\int_{-\infty}^{+\infty} e^{-(x-E_x)^2/(2E_x^2)}dx}{\sqrt{2\pi}E_n} = 1, \quad \Delta C \approx \mu_A(x) * \frac{\Delta x}{\left(\sqrt{2\pi}E_n\right)}. \tag{12}$$

Domain $(E_x - 3E_n, E_x + 3E_n)$, all elements in total contribution to the concept of

$$C_{E_x \pm 3E_n} \cdot C_{E_x \pm 3E_n} = \frac{1}{\sqrt{2\pi}E_n} \int_{E_x - 3E_n}^{E_x + 3E_n} \mu_A(x)\,dx = 99.74\%. \tag{13}$$

The contribution of different areas of the cloud droplet swarm on the qualitative concept is different in Figure 2.

X, E_n, and He^2 obey normal distribution. Probability density functions of E_n. Consider,

$$f_{E_n}(x) = \frac{1}{\sqrt{2\pi}\mathrm{He}} e^{(x-E_n)^2/2\mathrm{He}^2}, \quad f_x(x \mid E_n) = \frac{1}{\sqrt{2\pi}E_n'} e^{(x-Ex)^2/2E_n'^2}. \tag{14}$$

It can be inferred that

$$f(x) = \frac{1}{\sqrt{2\pi}E_n} e^{(x-E_x)^2/2E_n^2},$$

$$f_x(x) = f_{E_n'}(x) \times f_x(x \mid E_n) = \int_{-\infty}^{+\infty} \frac{1}{2\pi \mathrm{He}\,|y|} e^{((x-E_x)^2/2y^2)-((y-E_n)^2/2\mathrm{He}^2)} dy,$$

$$f_{X,\mu}(x) = f_\mu(y) f_X(x \mid \mu = y) = \begin{cases} \dfrac{1}{2\pi \mathrm{He} \ln y} e^{(x-E_x-\sqrt{-2\ln y}E_n)^2/4\mathrm{He}^2 \ln y}, \\ \quad (0 < y \le 1, E_x \le x < +\infty), \\ \dfrac{1}{2\pi \mathrm{He} \ln y} e^{(x-E_x+\sqrt{-2\ln y}E_n)^2/4\mathrm{He}^2 \ln y}, \\ \quad (0 < y \le 1, -\infty < x \le E_x). \end{cases} \tag{15}$$

We can calculate the digital characteristic entropy estimate values $\widehat{E}_n = (\overline{z}^2 - (S^2/2))^{1/4}$. At the same time, we can calculate the digital characteristic. The excess entropy estimate values $\widehat{H}_e = \sqrt{(\overline{z} - (\overline{z}^2 - (S^2/2))^{1/2}}$).

Cloud model is also used in two-order Gaussian distribution to produce cloud drops whose distribution displays high kurtosis and fat tail with power-law decay. In sociology and economics, many phenomena have been found to share high kurtosis and fat tail because of preferential attachment in evolution processes. This paper tries to investigate the relation between Gaussian distribution and fat-tail distribution, and to construct fat-tail distribution based on Gaussian distribution with iterations to depict more uncertain phenomena. If the Gaussian distribution as 1 order Gaussian distribution, and then order 1 x_1 probability density function is expressed as a random variable. Consider,

$$f(x) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right) e^{(x-u)^2/2\sigma^2}. \tag{16}$$

For three of the higher order Gaussian iteration with simple parameters, we study the changing trend of its mathematical properties and make lots of experimental analysis.

When $u_i = 0$ $(i = 1, 2, \ldots, n)$, $u_i = 0$ and $\sigma \neq 0$; then

$$E(X_i) = u_i = 0,$$
$$\mathrm{Var}(x_i) = \sum_{i=1}^{n-1} u_i^2 + \sigma^2 = \sigma^2,$$
$$\frac{E(x_i - E(x_i))^4}{\mathrm{Var}(x_i)} = \frac{3n\sigma^4}{\sigma^4}. \tag{17}$$

When $u_i = u$ $(i = 1, 2, \ldots, n)$ and $r = \sigma/u$; then

$$E(X_i) = u_i = u,$$
$$\mathrm{Var}(x_i) = \sum_{i=1}^{n-1} u_i^2 + \sigma^2 = \sigma^2 + (n-1)\frac{\sigma^2}{r^2},$$
$$\frac{E(x_i - E(x_i))^4}{\mathrm{Var}(x_i)} = \frac{3^p r^4 + 6r^2 \sum_{i=1}^{n-1} 3^{n-i} + \sum_{i=1}^{n-1} (6i-5) 3^{n-i}}{r^4 + (n-1)^2 + 2(n-1) r^2}. \tag{18}$$

When $u_i = r^i \sigma$ $(i = 1, 2, \ldots, n)$, then $E(X_i) = u_i = r^i \sigma$,

$$\mathrm{Var}(x_i) = \sum_{i=1}^{n-1} u_i^2 + \sigma^2 = \left(\sum_{i=1}^{n-1} r^{2i} + 1\right)\sigma^2 = \sigma^2\left(\frac{r^{2n}-1}{r^2-1}\right),$$
$$\frac{E(x_i - E(x_i))^4}{\mathrm{Var}(x_i)} = \left(3^p + 6\sum_{i=1}^{n-1} 3^{n-i} r^{2i} + 6\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} 3^{n-i} r^{2(i+j)} + \sum_{i=1}^{n-1} 3^{n-i} r^{4i}\right) \times \left(1 + 2\sum_{i=1}^{n-1} r^{2i} + \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} r^{2(i+j)}\right)^{-1}. \tag{19}$$

In short, the combinatorial clustering algorithms based on the Cloud, transform a qualitative question, the effectiveness evaluation of the combinatorial clustering algorithms into a quantitative question, and consider the fuzzy factors and random factors effects appeared in the evaluating cause. In the cloud model, E_n determines the steepness of the normal cloud, the greater the E_n, the wider the cloud covered the level range, otherwise the more narrow. According to "$3E_n$" rule of normal cloud, take $\theta_1 = 3$ in the algorithm, which affect the degree of discrete cloud droplets.

2.3. Combinatorial Optimization Clustering Algorithm. Combination with the basic principle of particle swarm optimization, a rapid evolutionary algorithm is proposed based on the characteristics of the cloud model on the process of transforming a qualitative concept to a set of quantitative numerical values, namely cloud hyper mutation particle swarm optimization algorithm. Its core idea is to achieve the evolution of the learning process and the mutation operation by the normal cloud particle operator. The COCQPSO can be modeled naturally and uniformly, which makes it easy and natural to control the scale of the searching space. The simulation results show that the proposed algorithm has fine capability of finding global optimum, especially for multimodal functions.

Table 1: Solutions of sphere functions with different dimensionality N values.

Dimensionality	$N = 2$	$N = 5$	$N = 10$	$N = 30$
5	$8.785e-165$	$1.650e-144$	$4.176e-129$	$4.666e-120$
10	$2.979e-129$	$1.237e-125$	$4.412e-117$	$5.301e-111$
30	$1.192e-067$	$3.888e-071$	$1.766e-068$	$3.069e-060$
50	$4.201e-049$	$3.038e-050$	$3.121e-048$	$1.191e-041$
100	$5.127e-029$	$6.976e-031$	$4.380e-030$	$7.082e-026$

Table 2: Dimension, initial value scope, and target values of 6 test functions.

Number	Functions	Dimension	The scope of initial values (10^4)	Acceptable level (%)
F1	Sphere	30	[−100, 100]	0.0100
F2	Rosenbrock	30	[−100, 100]	100.00
F3	Rastrigrin	30	[−100, 100]	100.00
F4	Griewank	30	[−600, 600]	0.1000
F5	Schaffer	2	[−100, 100]	0.00001
F6	Ackley	30	[−32, 32]	0.1000

Choose global extreme value Gbest in mutation as E_x, because at this point it may have been trapped in local optimal algorithm. And according to the principle of sociology, the current outstanding individuals usually also exist better individuals. There are two positions are used in the particle for updating the velocity; one is the global experience position of all particles, which memorizes the global best solution obtained through all particles; the other is each particle's local optimal point, which memorizes the best position that particle has ever moved to. Namely the local optimal point in its surroundings more chance to find the optimal solution. Parameter K is set too large and the variation is too frequent, which influence the operational efficiency of algorithm. Set is too small, the precision will be reduced. And because of the particle swarm algorithm in the early evolutionary convergence speed, convergence speed gradually slows down late, so it is difficult to give parameters K set a perfectly reasonable fixed values. In this paper, we make the $K = G_{\text{best}}/2$, the K value along with the global optimal value Gbest dynamic decreases, and realized adaptive adjustment.

For threshold selection of N, Sphere function, for example, to verify the influence of different N values of CHPSO algorithm precision. The experiment parameter settings are as follows: population sizes are 100, scope of the initial value for (−5, 5), the largest iterative algebra are 1, 000, respectively, for, 10, 30, and 50 dimension 5 and 100 Sphere function, in the case of N take 2, 5, 10, and 20 independent running 50 times averaging, measure parameter $K = G_{\text{best}}/2$. The experimental results are shown in Table 1.

We can see from Table 1, the smaller is the threshold N of dimensionality than 10 low dimensional function, the higher is the accuracy of the solution. For functions test, the acceptable result proportion of the value is shown in Table 2.

Test functions dimension, the range of initial value, the acceptable values of the experimental results of the COCQPSO algorithm, the Cloud genetic algorithm (CGA), the adaptive particle swarm optimization (APSO), Gaussian-Dynamic PSO, GDPSO), are shown in Table 3. All experimental population size is 20, the number of iterations is 1, 000, 100 times and each function operate independently and then we compare the optimal value, average value, and the success rate. To fully embody the effect of mutation operator, COCQPSO. Dynamically adjusted inertia weight w is not in, take a fixed value 0.725. Accelerate the constants $C_1 = 1$ and $C_2 = 1$, threshold value $N = 2$.

Use the knowledge of probability theory to prove the convergence of the COCQPSO algorithm. The combinatorial algorithm could be adaptive control under the guidance of the scope of the search space, and they are best under the conditions of the larger search space to avoid the local optimal solution. A typical functions of comparative experiment results show that the algorithm can avoid trapping in local optimal solution, and enhance the ability of global optimization at the same time to be able to more quickly converge to the global optimal solution. At the same time, the quality and efficiency of optimization is better than the other algorithms. The simulation results show that the complex constrained optimization problem, optimization algorithm, and excellent performance, especially for the super high-dimensional constraint optimization problem, the combinatorial algorithm obtained the higher accuracy of the solution.

3. Simulation Experiment Results

3.1. The Comparison Results of the Algorithms. PSO is inspired by social behavior among individuals, for instance, bird flocks. Particles (individuals) representing a potential problem solution move through the search space. The COCQPSO and FCOCQPSO algorithms focus on the collective behaviors that result from the local interactions of the individuals and interactions with their environment. The algorithms experimental results prove that the hybridization strategies are effective. The sequence algorithms with the enlarged pheromone table are superior to the other algorithms because the enlarged pheromone table diversifies the generation of

Table 3: Performance comparison among APSO, GDPSO, and COCQPSO.

Functions	APSO			GDPSO		
	Average	Success rate	Optimal value	Average	Success rate	Optimal value
Sphere	$1.43e-18$	100.00	$3.99e-29$	$1.21e-08$	100.00	$1.08e-25$
Rosenbrock	36.0900	95.00	0.0003	25.659	98.00	24.9651
Rastrigrin	65.4700	100.00	0.0000	17.1300	98.00	0.000
Griewank	0.00824	100.00	0.0000	$1.80e-03$	100.00	0.000
Schaffer	0.99951	98.00	0.0000	$6.06e-04$	74.00	0.000
Ackley	$1.12e-5$	100.00	—	—	—	—
Functions	CGA			COCQPSO		
	Average	Success rate	Optimal value	Average	Success rate	Optimal value
Sphere	$1.4949e-007$	100.00	$1.3147e-09$	$1.01e-10$	100.00	$4.2461e-225$
Rosenbrock	$25.749e-008$	98.00	$9.6253e-09$	26.4879	98.00	$9.0725e-008$
Rastrigrin	3.000000	100.00	3.000000	3.000000	100.00	3.000000
Griewank	0.0000	100.00	0.0000	0.0000	100.00	0.0000
Schaffer	−1.031628	100.00	−1.031628	−1.03156	100.00	−1.031601
Ackley	0.998004	100.00	0.998004	$3.90e-12$	100.00	$4.44e-15$

Table 4: Comparison of error rates for the algorithms.

Data set	Criteria	GA	ACO	PSO	COCQPSO	FCOCQPSO
Data 1	Best	2.9230	2.6720	2.9220	2.8520	2.8500
	Worst	3.0000	2.7030	3.0000	2.9300	2.9282
	Average	**2.9600**	**2.6900**	**2.9510**	**2.8810**	**2.8800**
Data 2	Best	0.3440	1.5780	0.3430	0.2740	0.2730
	Worst	0.3760	1.6100	0.3750	0.3060	0.3045
	Average	**0.3600**	**1.5940**	**0.3590**	**0.2895**	**0.2890**
Data 3	Best	3.3380	2.9840	3.3280	2.6300	2.6100
	Worst	3.3850	3.0470	3.3750	3.3000	3.2980
	Average	**3.3750**	**3.0155**	**3.3650**	**3.3020**	**3.3000**
Data 4	Best	11.8530	18.7340	11.8430	11.1050	11.0045
	Worst	11.9160	18.9060	11.9060	11.2040	11.1800
	Average	**11.8730**	**18.8150**	**11.8630**	**11.1500**	**11.1020**

new solutions of the COCQPSO and FCOCQPSO algorithms, which prevent traps into the local optimum.

To compare the performance of the COCQPSO and FCOCQPSO algorithms with those of other clustering algorithms such as Genetic algorithm (GA) and ant colony optimization (ACO), we have made lots of clustering experiments such as four different types of artificial data sets to get the average and standard deviation. Four different types of artificial data sets have been randomly generated from a multivariate uniform distribution. Data set 1: the wine dataset. This dataset contains chemical analyses of 178 wines derived from three different cultivars. Data set 2: the iris dataset. This dataset contains three categories of 50 objects. Data set 3: the glass identification dataset. This dataset contains 214 objects with nine attributes. Data set 4: the CMC dataset. Clustering experimental results of comparison of error rates for the algorithms is given in Table 4.

From the results given in Table 4, we can see that none of the previously proposed algorithms outperform the COCQPSO and FCOCQPSO algorithms in terms of the average and best objective function values for the four datasets used here. The proposed algorithms of this study are effective, robust, easy to tune, and tolerably efficient as compared with other algorithms.

The COCQPSO and FCOCQPSO algorithms have been tested in comparison with the GA and PSO for artificial and real world data. Finally, good experimental results show the feasibility of the proposed algorithms. In all our experiments, both the data is randomly or heuristically initialized. For complex problems, it is advised to choose the FCOCQPSO algorithm for the initialization, in order to reach the best solutions. Clustering results of the FCOCQPSO algorithm are shown in Figure 3.

To sum up, simulation experiment results show a superiority of the FCOCQPSO algorithms over other counterpart algorithms in terms of accuracy. In the COCQPSO and FCOCQPSO algorithms, the particle's search is guided by the position which may be not the global position but may lie in a promising search region so that the particles have a big chance to search this region and find out the global optimal solution.

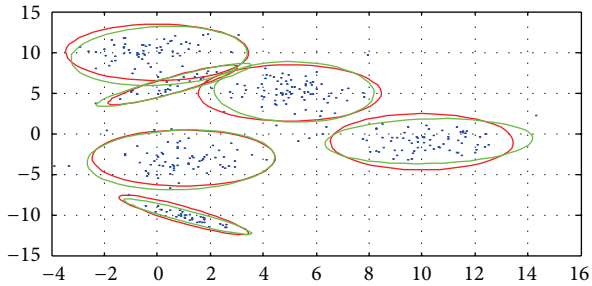


Figure 3: Clustering results of the FCOCQPSO algorithm.

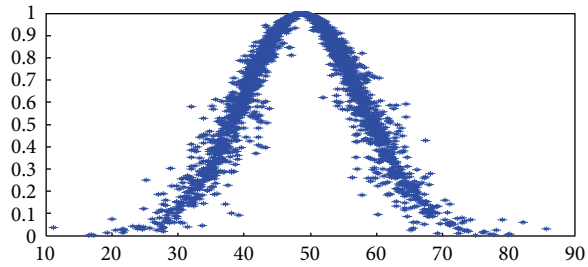


Figure 4: He = 1.89 clustering algorithm.

As a result, the COCQPSO and FCOCQPSO algorithms may have stronger global search ability and better overall performance than the other algorithms such as GA, PSO, ACO, and DE, particularly for the hard optimization problems. The experimental results show that the combinatorial clustering algorithms improve the precision and the result is stable.

3.2. Experimental Results. Fuzzy particle swarm optimization clustering algorithm is a novel method for solving real problems by using both the fuzzy rules and the characteristics of particle swarm optimization. In this paper, we successfully solve multiattribute groups' allocation problems by fuzzy combinatorial optimization clustering algorithm of cloud model and quantum-behaved particle swarm optimization.

Hype Entropy H_e reflects the dispersion of the Cloud drops. The bigger the Hyper Entropy is, the bigger of its dispersion and the randomness of degree of membership, and so is the thickness of Cloud. Two figures contrast can be seen, big super entropy diagram cloud droplets of discrete degree are big, and the entropy of small cloud droplets is relatively concentrated. The clustering algorithm result of the cloud droplets experiments phenomenon is shown in Figure 4.

Fuzzy clustering algorithm of data to some extent, overcomes the inner shape: the distribution of the dependent on, and can correctly in different shape distribute data clustering, compute speed fast, and overcome the sensitivity to noise data and outliers, enhancing the algorithm robustness. According to fuzzy clustering, algorithm is sensitive to initial value, the shortcomings of easy to fall into local optimum, this paper proposes a method of fuzzy clustering based on particle swarm optimization. The design of fitness function according to the clustering criterion, using particle swarm optimization

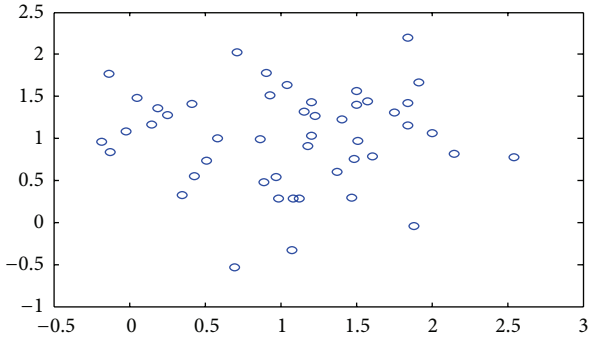


Figure 5: Combinatorial optimization clustering algorithm result.

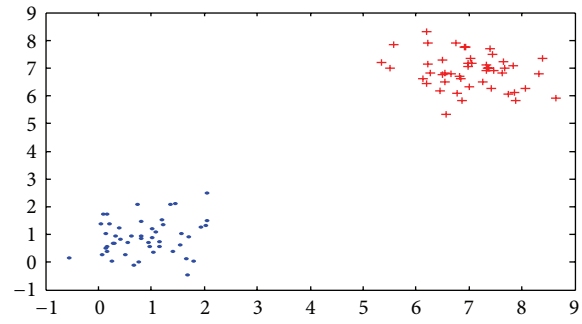


Figure 6: The corresponding dynamic clustering algorithm result.

algorithm to optimize clustering center, behind the simulation experiment proves the feasibility and effectiveness of the algorithm.

According to different thresholds of multiproject, multiattribute groups, we get the corresponding dynamic cluster result by algorithm of cloud model and quantum-behaved particle swarm optimization as follows in Figure 5.

According to different thresholds of multiproject, multiattribute groups, we get the corresponding dynamic cluster result by combinatorial optimization algorithm of cloud model and quantum-behaved particle swarm optimization clustering algorithm as follows in Figure 6.

According to different thresholds of multiproject, multiattribute groups, we get the corresponding 200-dynamic combinatorial cluster result as follows in Figure 7.

According to different thresholds of multiproject, multiattribute groups, we get the corresponding dynamic combinatorial cluster result by 200-quantum-behaved particle swarm optimization and cloud model clustering algorithm as follows in Figure 8.

According to different thresholds of multiproject, multiattribute groups we get the corresponding 1000-dynamic combinatorial cluster result as follows in Figure 9.

According to different thresholds of multiproject, multiattribute groups, we get the corresponding dynamic cluster result by 1000-quantum-behaved particle swarm optimization and cloud model combinatorial clustering algorithm as follows in Figure 10.

According to different thresholds of multiproject, multiattribute groups, we get the corresponding 10000-dynamic

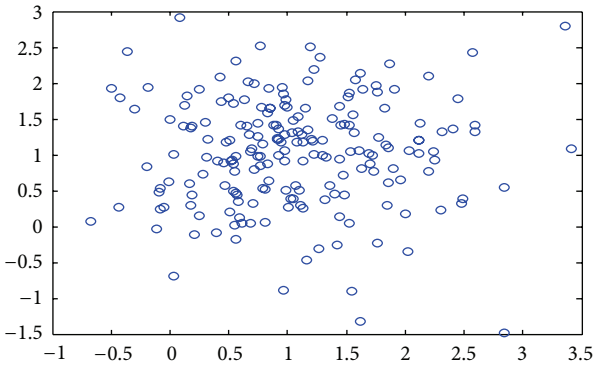


Figure 7: 200-dynamic clustering algorithm result.

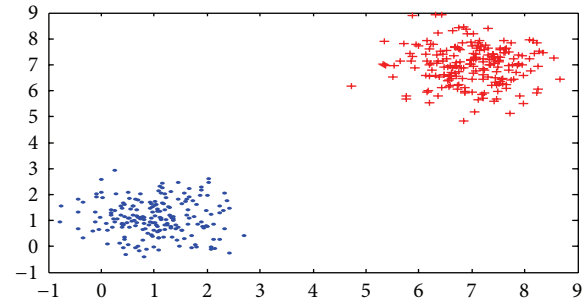


Figure 8: 200-valued combinatorial clustering algorithm result.

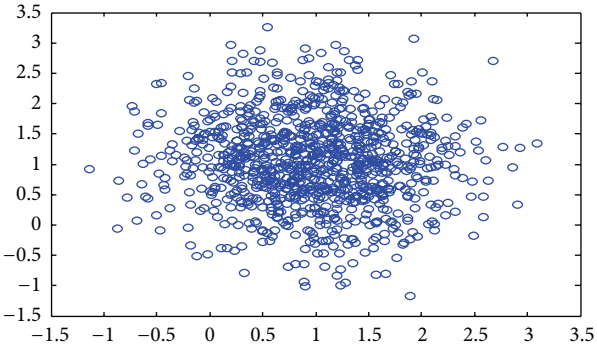


Figure 9: 200-dynamic clustering algorithm result.

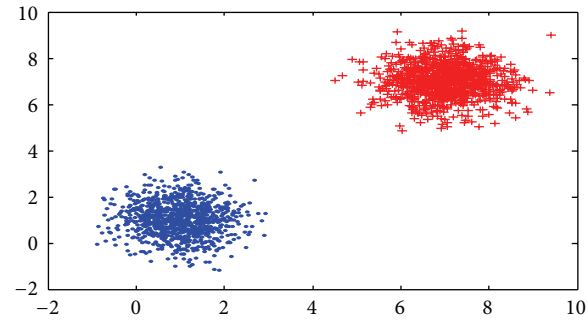


Figure 10: 1000-valued combinatorial clustering algorithm result.

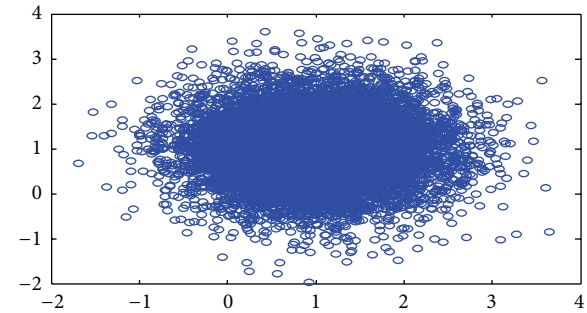


Figure 11: 10000-dynamic clustering algorithm result.

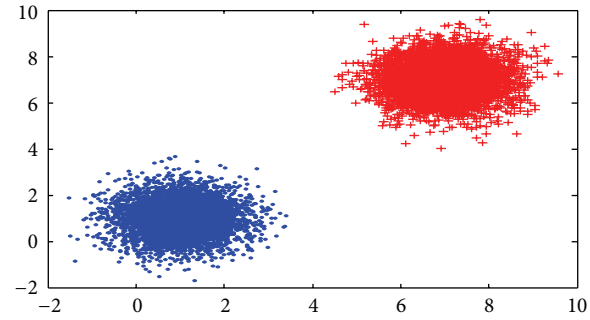


Figure 12: 10000-valued combinatorial clustering algorithm cluster result.

combinatorial cluster result by quantum-behaved particle swarm optimization clustering algorithm as follows in Figure 11.

According to different thresholds of multiproject, multiattribute groups, we get the corresponding dynamic cluster result by 10000-quantum-behaved particle swarm optimization and cloud model combinatorial clustering algorithm as follows in Figure 12.

4. Conclusion

According to these studies, there are many different kinds of clustering algorithms which can be applied. In addition, it is critical to have a suitable algorithm for the effectiveness of a clustering method. In this paper, we have developed the COCQPSO and FCOCQPSO algorithms. The results from various simulations using artificial data sets show that the proposed combinatorial clustering algorithms have better performance than that of the other clustering algorithms such as GA, PSO, and ACO. We demonstrated through several experiments that much better clustering results can be got by interval-valued combinatorial clustering algorithm than the corresponding dynamic clustering algorithm.

These experiments show that the combinatorial clustering algorithm based on similarity is better than conventional clustering method in terms of calculating complexity and clustering effect. In this work, we have given the full description of implementations and details of the combinatorial algorithms to improve its performance. We have tested the proposed algorithms in different synthetic and real clustering

problem, obtaining very good results that improve classical approaches. So, it is very important for us to research on combinatorial clustering algorithms. The combinatorial clustering algorithms are applied in several fields such as statistics, pattern recognition, machine learning, and data mining, to partition a given set of data or objects into clusters. It is also applied in a large variety of applications, for example, image segmentation, objects and character recognition, and document retrieval. The combinatorial clustering algorithms problems are the kind of important optimization problems in our real life which are difficult to find a satisfying solution within a limited time. In addition to the extension of the application domain, our future studies can investigate the combinatorial clustering algorithms to improve their solution quality.

Acknowledgments

This research is partially supported by the National Natural Science Foundation of China (Grant no. 70372039, Grant no. 70671037, and Grant no. 70971036) and the Doctoral Program Foundation of Institutions of Higher Education of China (Grant no. 20050532005).

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 316–323, 1999.

[2] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-means clustering," *Pattern Recognition Letters*, vol. 25, no. 11, pp. 1293–1302, 2004.

[3] S. Kotsiantis and P. Pintelas, "Recent advances in clustering: a brief survey," *WSEAS Transactions on Information Science and Applications*, vol. 1, no. 1, pp. 73–81, 2004.

[4] J.-S. Lee and S. Olafsson, "Data clustering by minimizing disconnectivity," *Information Sciences*, vol. 181, no. 4, pp. 732–746, 2011.

[5] H. Alizadeh, B. Minaei-Bidgoli, and S. K. Amirgholipour, "A new method for improving the performance of k nearest neighbor using clustering technique," *Journal of Convergence Information Technology*, vol. 4, no. 2, pp. 84–92, 2009.

[6] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[7] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 450–465, 1999.

[8] Y. Leung, J.-S. Zhang, and Z.-B. Xu, "Clustering by scale-space filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1396–1410, 2000.

[9] E. Forgy, "Cluster analysis of multivariate data: efficiency vs. interpretability of classification," *Biometrics*, vol. 21, pp. 768–780, 1965.

[10] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, John Wiley & Sons, New York, NY, USA, 1990.

[11] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773–780, 1989.

[12] A. Lorette, X. Descombes, and J. Zerubia, "Fully unsupervised fuzzy clustering with entropy criterion," in *International Conference on Pattern Recognition*, vol. 3, pp. 3998–4001, 2000.

[13] K.-Y. Huang, "A synergistic automatic clustering technique (SYNETRACT) for multispectral image analysis," *Photogrammetric Engineering and Remote Sensing*, vol. 68, no. 1, pp. 33–40, 2002.

[14] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data," *Behavioral Science*, vol. 12, no. 2, pp. 153–155, 1967.

[15] D. Pelleg and A. Moore, "X-means: extending K-means with efficient estimation of the number of clusters," in *Proceedings of the 17th International Conference on Machine Learning*, pp. 727–734, Morgan Kaufmann, San Francisco, Calif, USA, 2000.

[16] G. Hamerly and C. Elkan, "Learning the K in K-means," in *Proceedings of 7th Annual Conference on Neural Information Processing Systems (NIPS '03)*, pp. 281–288, 2003.

[17] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Analysis and Applications*, vol. 8, no. 4, pp. 332–344, 2006.

[18] S. Ouadfel, M. Batouche, and A. Taleb-Ahmed, "A new particle swarm optimization algorithm for dynamic image clustering," in *Proceedings of the 5th International Conference on Digital Information Management (ICDIM '10)*, pp. 546–551, July 2010.

[19] C. S. Wallace and D. M. Boulton, "An information measure for classification," *Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968.

[20] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Pearson Education, 2006.

[21] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[22] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, IEEE Press, Piscataway, NJ, USA, May 1998.

[23] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of IEEE Congress on Evolutionary Computation (ICEC '99)*, pp. 1951–1957, Washington, DC, USA, 1999.

[24] F. Van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimiser," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 94–99, October 2002.

[25] B. Zhao, C. X. Guo, B. R. Bai, and Y. J. Cao, "An improved particle swarm optimization algorithm for unit commitment," *International Journal of Electrical Power and Energy Systems*, vol. 28, no. 7, pp. 482–490, 2006.

[26] M. S. Arumugam and M. V. C. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 324–336, 2008.

[27] M. A. Montes de Oca, T. Stützle, M. Birattari, and M. Dorigo, "Frankenstein's PSO: a composite particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120–1132, 2009.

[28] J. Kennedy and R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4104–4108, IEEE Service Center, Piscataway, NJ, USA, October 1997.

[29] L. Wang and J. Yu, "Fault feature selection based on modified binary PSO with mutation and its application in chemical process fault diagnosis," in *Proceedings of the 1st International Conference on Advances in Natural Computation (ICNC '05)*, vol. 3612 of *Lecture Notes in Computer Science*, pp. 832–840, Changsha, China, August 2005.

[30] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–37, 2008.

[31] X. F. Xie, W. J. Zhang, and Z. L. Yang, "A dissipative particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computing (CEC '02)*, pp. 1666–1670, Honolulu, Hawaii, USA, 2002.

[32] L. Wang, L. Zhang, and D.-Z. Zheng, "An effective hybrid genetic algorithm for flow shop scheduling with limited buffers," *Computers & Operations Research*, vol. 33, no. 10, pp. 2960–2971, 2006.

[33] L. Nanni and A. Lumini, "Particle swarm optimization for prototype reduction," *Neurocomputing*, vol. 72, no. 4-6, pp. 1092–1097, 2009.

[34] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Fractional particle swarm optimization in multidimensional search space," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 2, pp. 298–319, 2010.

[35] S. Ouadfel, M. Batouche, and A. Taleb-Ahmed, "A new particle swarm optimization algorithm for dynamic image clustering," in *Proceedings of the 5th International Conference on Digital Information Management (ICDIM '10)*, pp. 546–551, July 2010.

[36] E. Martinez, M. M. Alvarez, and V. Trevino, "Compact cancer biomarkers discovery using a swarm intelligence feature selection algorithm," *Computational Biology and Chemistry*, vol. 34, no. 4, pp. 244–250, 2010.

[37] Y.-T. Kao, E. Zahara, and I.-W. Kao, "A hybridized approach to data clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1754–1762, 2008.

[38] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, and F. C. Tien, "Integration of particle swarm optimization and genetic algorithm for dynamic clustering," *Information Sciences*, vol. 195, pp. 124–140, 2012.

[39] H. Cheng, S. Yang, and J. Cao, "Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc networks," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1381–1392, 2013.

[40] D. Chang, Y. Zhao, C. Zheng, and X. Zhang, "A genetic clustering algorithm using a message-based similarity measure," *Expert Systems with Applications*, vol. 39, no. 2, pp. 2194–2202, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 531031, 13 pages
http://dx.doi.org/10.1155/2013/531031

Research Article

Artificial Hydrocarbon Networks Fuzzy Inference System

Hiram Ponce, Pedro Ponce, and Arturo Molina

Graduate School of Engineering, Tecnológico de Monterrey, Campus Ciudad de México, 14380 Mexico City, DF, Mexico

Correspondence should be addressed to Hiram Ponce; hiram.eredin@gmail.com

Received 13 May 2013; Revised 25 July 2013; Accepted 1 August 2013

Academic Editor: Chen

This paper presents a novel fuzzy inference model based on artificial hydrocarbon networks, a computational algorithm for modeling problems based on chemical hydrocarbon compounds. In particular, the proposed fuzzy-molecular inference model (FIM-model) uses molecular units of information to partition the output space in the defuzzification step. Moreover, these molecules are linguistic units that can be partially understandable due to the organized structure of the topology and metadata parameters involved in artificial hydrocarbon networks. In addition, a position controller for a direct current (DC) motor was implemented using the proposed FIM-model in type-1 and type-2 fuzzy inference systems. Experimental results demonstrate that the fuzzy-molecular inference model can be applied as an alternative of type-2 Mamdani's fuzzy control systems because the set of molecular units can deal with dynamic uncertainties mostly present in real-world control applications.

1. Introduction

It is well known that fuzzy inference models are very important in applications when information is uncertain and imprecise, like: robotics, medicine, control, modeling, and so forth [1–6]. Moreover, fuzzy inference models may deal with nonlinearities in the input domain to transform them into an output domain. In that way, the literature reports three main models: Takagi-Sugeno inference systems [7], Mamdani's fuzzy control systems [8], and Tsukamoto's inference model [9].

Roughly speaking, Takagi-Sugeno inference systems apply polynomial functions to construct the consequent values using pairs of input-output data of a given system to model [7]. Mamdani's fuzzy control systems refer to control laws that apply fuzzy inference models with fuzzy partitions in the defuzzification phase [8], obtaining mostly the output value with the center of gravity (COG) function [10]. In contrast, Tsukamoto's inference models implement monotonic membership functions [9]. For detailed information, see [11].

The above inference models were developed under type-1 fuzzy systems. However, these models have disadvantage in terms of dynamic uncertainties present at inputs. For example, the latter gives poor performance in control systems because real-world control applications present dynamic uncertainties inherently [12, 13]. In contrast, type-2 fuzzy systems were proposed as an improvement of type-1 fuzzy inference systems. For instance, recent applications on fuzzy control systems have demonstrated the ability of type-2 fuzzy control systems to handle with noise and perturbations [12–14].

On the other hand, other fuzzy inference models have been proposed as hybrid algorithms using heuristics to manage unusual information, pattern recognition, and learning. Some of these fuzzy inference models use genetic algorithms, harmony search algorithms, tabu search, artificial neural networks, swarm intelligence techniques, and so forth [3, 4, 15].

Recently, H. Ponce and P. Ponce [16–20] proposed a new computational algorithm for modeling problems named artificial hydrocarbon networks based on natural hydrocarbon compounds. This algorithm claims for stability, well forming of compounds, easiness of spanning structures, and a degree of interpretation of the resultant model based on organized structures. In particular, the basic unit of information in this algorithm is the molecule. Actually, molecules are simple packages of information that can be inherited and interpreted. At last, basic chemical rules are applied to build the final structure.

Then, the objective of this paper is to present a novel fuzzy inference model based on artificial hydrocarbon

networks named fuzzy-molecular inference (FIM) model. In that sense, molecules can model consequent values of fuzzy rules and partition linguistic variables. Moreover, a fuzzy control system based on the FIM-model is presented as a case study. Experimental results demonstrate that the fuzzy-molecular inference model can be applied as an alternative of type-2 Mamdani's fuzzy control systems because the set of molecular units can deal with dynamic uncertainties mostly present in real-world control applications.

The paper is ordered as follows. Next section presents a review of artificial hydrocarbon networks algorithm introduced recently in [16–20]. The following sections introduce new material. Section 3 describes the fuzzy-molecular inference model in detail, current proposal of the paper. Section 4 introduces an example of how to apply the FIM-model in fuzzy control systems. Section 5 presents a case study in which a type-2 fuzzy control system based on the FIM-model implements a position controller of a direct current (DC) motor. Section 6 presents the experimental results of the case study and discusses some differences between the proposed model and other fuzzy inference systems and the advantages of the FIM-model to be used as an alternative of type-2 fuzzy systems. Finally, Section 7 concludes the paper and presents future directions.

2. Artificial Hydrocarbon Networks

In this section, a brief review of artificial hydrocarbon networks is presented. However, this algorithm is subjected to the artificial organic networks technique. Thus, a first description of artificial organic networks technique is discussed and then artificial hydrocarbon networks algorithm is formally introduced.

2.1. Brief Review of Artificial Organic Networks. Observations to chemical organic compounds reveal enough information to derive the *artificial organic networks* technique firstly proposed by H. Ponce and P. Ponce [16–22]. From studies of organic chemistry, organic compounds are the most stable ones in nature. In addition, molecules can be seen as units of packaging information; thus, complex molecules and its combinations can determine a nonlinear interaction of information. Moreover, molecules can be used for encapsulation and potential inheritance of information. Thus, artificial organic networks take advantage of this knowledge, inspiring a computational algorithm that infer and classify information based on stability and chemical rules that allow formation of molecules [19, 21].

Artificial organic networks (AONs for short) define four components: atoms, molecules, compounds, and mixtures; and two basic interactions among components: covalent bonds and chemical balance interaction. In order to follow chemical rules, the following definitions of AONs hold [16–22].

(a) Atoms. They are the basic units with structure. No information is stored. In addition, when two atoms have the same number of degrees of freedom they are called *similar atoms* and *different atoms* otherwise. The degree of freedom is the number of valence electrons that allow atoms to be linked with others.

(b) Molecules. They are the interactions of two or more atoms made of covalent bonds. These components have structural and behavioral properties. Structurally, they conform the basis of an organized structure while behaviorally they can contain information. Thus, molecules are known as the basic units of information. If a molecule has filled out all of the valence electrons in atoms, it is *stable*; but if a molecule has at least one valence electron without filling, it is considered as *unstable*.

(c) Compounds. In structure, they are two or more molecules interacting with each other linked with covalent bonds. Their behaviors are mappings from the set of molecular behaviors to real values.

(d) Mixtures. They are the interaction of two or more molecules and/or compounds without physical bonds. Mixtures are linear combinations of molecules and/or compounds forming a basis of molecules with weights so-called *stoichiometric coefficients*.

(e) Covalent Bonds. They are of two types. For this work, *polar covalent bonds* refer to the interaction of two similar atoms, while *nonpolar covalent bonds* refer to the interaction of two different atoms.

(f) Chemical Balance Interaction. It refers to find the proper values of stoichiometric coefficients in mixtures in order to satisfy constrains in artificial organic networks.

In fact, artificial organic networks follow the energy model [20] that states the hierarchical order in which components are used to form the final structure to minimize energy. For instance, the first strategy considers formation of molecules. If molecules cannot deal with the problem, compounds are required. Finally, mixtures of molecules and/or compounds will act.

2.2. Description of Artificial Hydrocarbon Networks. Artificial hydrocarbon networks (AHNs for short) algorithm is based on artificial organic networks that implement notions of natural hydrocarbon compounds [19, 21]. Formally, AHNs define components, interactions, and the training algorithm, in order to infer and classify information given any system. In that way, two main procedures are needed for AHNs: training and reasoning. Following, a brief review of artificial hydrocarbon networks is presented.

2.2.1. Basic Components. In particular to AHNs, only two types of atoms are considered: hydrogen atoms H and carbon atoms C. Those have valence electrons $e_H = 1$ and $e_C = 4$ for the hydrogen and carbon atoms, respectively. In that sense, hydrocarbon atoms can be bonded with at most one atom while carbon atoms can be bonded with at most four, knowing as the octet rule [16–22].

The basic unit of information is a CH-molecule. These kinds of molecules are structurally made of hydrogen and carbon atoms following the octet rule. Let M_i be the structure of a molecule, and, φ_i be the behavior of molecule M_i. Then, φ_i is a mapping from some set X to real numbers $\mathbb{R}$. Moreover, let M_H and M_C be two molecules with behaviors φ_H and φ_C, and if (1) holds for these behaviors, then M_H and M_C are called CH-molecules, where h is complex constant value named hydrogen value, x is any input value $|x| \leq 1$ that excites a molecule, and d is the number of hydrogen atoms attached to a carbon atom:

$$\varphi_H(x) = h, \quad h \in \mathbb{C},$$
$$\varphi_C(x) = \prod_{i=1}^{d \leq e_C} (x - \varphi_{Hi}), \quad x \in \mathbb{R}. \tag{1}$$

Let M_{CH}, M_{CH_2}, and M_{CH_3} be CH-molecules with behaviors φ_{CH}, φ_{CH_2}, and φ_{CH_3} like (2), respectively. Then, they are called CH-primitive molecules. Intuitively, these molecules can be seen as basic packages to join among them forming complex molecules-like compounds:

$$\varphi_{CH}(x) = (x - h_1),$$
$$\varphi_{CH_2}(x) = (x - h_1)(x - h_2), \tag{2}$$
$$\varphi_{CH_3}(x) = (x - h_1)(x - h_2)(x - h_3).$$

Let C_i be a compound formed with a set of p CH-molecules $\mathscr{M}$, and let ψ_i be the behavior of C_i. Then, ψ_i is expressed as (3), where φ_j are the behaviors of CH-molecules in $\mathscr{M}$ and π is the behavior of nonpolar covalent bonds that links molecules

$$\psi_i = \pi(\varphi_1, \ldots, \varphi_j, \ldots, \varphi_p, x). \tag{3}$$

Finally, let C_i be a n molecules or compounds with behaviors ψ_i. Then, S is a mixture of molecules or compounds and it is expressed as a linear combination of them like (4); where, α_i is a set of real values named stoichiometric coefficients representing the ratio of molecules or compounds occupied in the mixture.

$$S(x) = \sum_i \alpha_i \psi_i(x). \tag{4}$$

Let AHN be a mixture of molecules or compounds in the set Γ representing the structure of molecules or compounds (how they are connected), and, S be the behavior of the mixture. Then, AHN is called an artificial hydrocarbon network if Γ is spanned from CH-molecules. Figure 1 shows a simple artificial hydrocarbon network. It is remarkable to say that topology Γ is a fixed structure parameterized with hydrogen values h and stoichiometric coefficients α_i.

2.2.2. Training of Artificial Hydrocarbon Networks. Artificial hydrocarbon networks can deal with modeling problems like inferring or clustering in order to approximate any given system Σ with a pair of samples (x, y). In fact, let Σ be a simple-input-simple-output (SISO) system with input signal

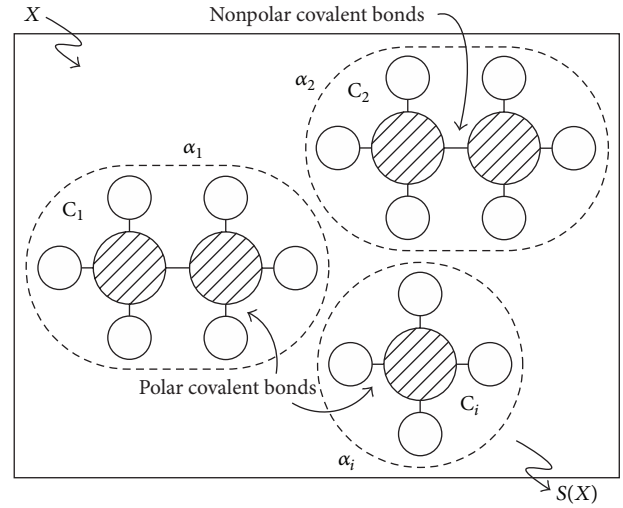


Figure 1: Simple artificial hydrocarbon network. *White circles* represent hydrogen atoms, and *black circles* represent carbon atoms.

x and output signal y. The training process of an artificial hydrocarbon network is summarized in Algorithm 1 which receives the sample pairs of Σ, the number of CH-molecules p and the number of compounds c. Algorithm 1 outputs the structure Γ, hydrogen values h, and stoichiometric coefficients α_i.

This is a modified algorithm from the original one reported in [21]. For instance, r^{jk} refers to an intermolecular distance which defines the distance between the position of two molecules M_j and M_k. Actually, the algorithm iteratively updates the set of intermolecular distances to define the best positions of molecules in the input domain using (5). In that sense, molecules will act under regions defined by these intermolecular distances. It is remarkable to say that the first molecule acts from the initial value of the input domain. In order to iteratively updates intermolecular distances, η is considered the step size or the learning rate, such that, $0 < \eta < 1$ and the least squares errors E_j and E_k for each molecule

$$r_{t+1}^{jk} = r_t^{jk} - \eta(E_j - E_k). \tag{5}$$

On the other hand, the original algorithm considers a generic interaction of CH-molecules referring to as a nonpolar covalent bond based training [20]. In this work, a linear chain of CH-molecules is adopted. Thus, each compound has a topology in the form of (6), where, the outside of the chain has M_{CH_3} molecules and M_{CH_2}; otherwise

$$M_{CH_3}\text{–}M_{CH_2}\text{–}\cdots\text{–}M_{CH_2}\text{–}\cdots\text{–}M_{CH_2}\text{–}M_{CH_3}. \tag{6}$$

Finally, Algorithm 1 considers adjustment parameters σ_j constant gain for molecular behaviors φ_j since φ_C in (1) is a normalized product form of a polynomial used in the least squares estimates (LSEs) method. In fact, consider the equivalence (7) when reasoning with AHNs. Where the set

(1) Initialize AHN = ∅
(2) **For each** C_i **do**
(3) Initialize r^{jk} randomly under the input domain
(4) **While** stop condition is not reached **do**
(5) Split (x, y)-pairs into p clusters y_j using r^{jk}
(6) **For each** cluster y_j **do**
(7) Create a CH-molecule using criterion (6)
(8) Obtain hydrogen values of molecule M_j using LSE method
(9) Calculate least square error E_j between y_j and M_j
(10) **End for**
(11) Update intermolecular distances r^{jk} using (5)
(12) **End while**
(13) Update AHN ← AHN ∪ C_i
(14) Update (x, y)-pairs with $(x, y -$ AHN$)$-pairs
(15) **End for**
(16) Obtain stoichiometric coefficients of C_i compounds using LSE method
(17) Return AHN

Algorithm 1: Training algorithm for artificial hydrocarbon networks.

of p values are coefficients of the polynomial form of φ_C of grade $d \leq e_C$

$$\sigma\varphi_C(x) = p_d x^d + p_{d-1}x^{d-1} + \cdots + p_1 x + p_0 = \sigma \prod_{i=1}^{d \leq e_C} (x - \varphi_{Hi}), \quad x \in \mathbb{R}. \tag{7}$$

2.2.3. Reasoning of Artificial Hydrocarbon Networks. Once the training is done, an artificial hydrocarbon network can be used for reasoning. In that sense, consider an input value x_0. The AHN has to be evaluated in x_0; thus, the reasoning value y_0 can be calculated using (8), where S is the behavior of the artificial hydrocarbon network, α_i are the stoichiometric coefficients, R is the set of all intermolecular distances between molecules, and σ_j are the adjustment parameters

$$y_0 = S(x_0 \mid H, \alpha_i, R, \sigma_j). \tag{8}$$

Notice that, if $c = 1$, it means that there exists one stoichiometric coefficient $\alpha_1 = 1$.

3. Description of the Proposed Fuzzy-Molecular Inference Model

The fuzzy-molecular inference model (FMI-model for short) is a fuzzy inference system that uses a fuzzy partition of input space in premises and artificial hydrocarbon networks in consequences as part of fuzzy implications. In this section, a detailed description of the fuzzy-molecular inference model is presented. For simplicity, through this section consider the FMI-model as a type-1 fuzzy system. In Section 5, an extension to type-2 fuzzy systems is presented.

Let A be a fuzzy set and its corresponding membership function $\mu_A(x)$ of A, for all $x \in X$, where X is the input domain space. In fact, the membership function is a value between 0 and 1 for representing the value of belonging x to the fuzzy set A.

Also, let R_i be the ith fuzzy rule of form as (9), where $\{x_1, \ldots, x_k\}$ is the set of variables in the antecedent, $\{A_1, \ldots, A_k\}$ is the set of the fuzzy partition of input space, y_i is the variable of the consequent, M_j is the jth CH-molecule of the artificial hydrocarbon network excited by the fuzzy implication process (see Section 3.3), and Δ is any T-norm function

$$R_i\text{: If } \Delta(x_1 \text{ is } A_1, \ldots, x_k \text{ is } A_k), \text{ then } y_i \text{ is } M_j. \tag{9}$$

If assuming that $\mu_\Delta(x_1, \ldots, x_k)$ is the result of the T-norm function as (10) with conjunction operator $\wedge$, then (9) can be rewritten as (11), where φ_j is the molecular behavior of M_j

$$\mu_\Delta(x_1, \ldots, x_k) = \mu_{A_1}(x_1) \wedge \cdots \wedge \mu_{A_k}(x_k), \tag{10}$$

$$R_i\text{: If } \Delta(x_1 \text{is } A_1, \ldots, x_k \text{ is } A_k), \text{ then } y_i = \varphi_j(\mu_\Delta(x_1, \ldots, x_k)). \tag{11}$$

Thus, the fuzzy-molecular inference model is finally expressed in (11). Figure 2 shows the fuzzy-molecular inference model as a block diagram. This model represents a nonlinear inference system for a given crisp input $x \in X$ that follows three steps, that is, *fuzzification*, *fuzzy inference engine*, and *defuzzification*, and obtains the corresponding crisp output $y \in Y$, where Y represents the output. Moreover, fuzzy rules like (9) can also be expressed as a fuzzy matrix that defines a *knowledge base* of the problem domain. Each block in the FMI-model is detailed in the following subsections.

3.1. Fuzzification. The fuzzy-molecular inference model can be viewed as a block with inputs and outputs. Moreover, let any given system be a single-input-single-output. Then, fuzzification maps any given input variable x, also known

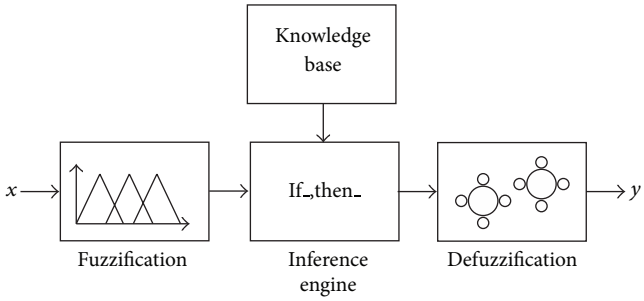


Figure 2: Block diagram of the fuzzy-molecular inference model.

as a linguistic variable, to a fuzzy value in the range $[0, 1]$. In particular, this mapping occurs using a fuzzy set A and its corresponding membership function $\mu_A(x)$, such that (12) holds:

$$\mu_A : x \longmapsto [0, 1]. \tag{12}$$

In fact, this linguistic variable is partitioned into m different fuzzy sets $\{A_i\}$, for all $i = 1, \ldots, m$. For example, this fuzzy partition can be "low," "medium," "high." Then, the evaluation of a given value of x is calculated using the set of membership functions $\mu_{A_i}(x)$, for all $i = 1, \ldots, m$.

The shape of all membership functions depends on the purpose of the problem domain. The literature reports different criteria and methods to do so as in [7–9, 11, 23, 24].

3.2. Fuzzy Inference Engine. Once the crisp value of the input is mapped to a fuzzy subspace as described in Section 3.1, the next step in the fuzzy-molecular inference model is the evaluation of the antecedents in fuzzy rules like (11). In this work, the min function (13) is selected for the T-norm Δ

$$\mu_\Delta(x_1, \ldots, x_k) = \min\left\{\mu_{A_1}(x_1), \ldots, \mu_{A_k}(x_k)\right\}. \tag{13}$$

Finally, the consequent value y_i is equal to the valued-behavior φ_j of the jth CH-molecule of an artificial hydrocarbon network. Thus, the consequent value y_i can be calculated using fuzzy rules (11) with the min function (13), as shown in (14)

$$y_i = \varphi_j\left(\min\left\{\mu_{A_1}(x_1), \ldots, \mu_{A_k}(x_k)\right\}\right). \tag{14}$$

3.3. Defuzzification. The last step in fuzzy-molecular inference model calculates the crisp value of the output y (15) using n fuzzy rules, where y_i is the consequent value and $\mu_{\Delta i}(x_1, \ldots, x_k)$ is the fuzzy evaluation of the antecedents, for $i = 1, \ldots, n$. In particular, (15) is based on the well-known center of gravity [10]

$$y = \frac{\sum \mu_{\Delta i}(x_1, \ldots, x_k) \cdot y_i}{\sum \mu_{\Delta i}(x_1, \ldots, x_k)}. \tag{15}$$

As noticed in Section 3.2, the fuzzy-molecular inference model requires a set of CH-molecules. In this case, let AHN be an artificial hydrocarbon network with one compound C that is made of p CH-primitive molecules with molecular behavior of the form as in (2). In this work, compound C is restricted to a linear chain of CH-molecules like in (16), where – stands for a covalent bond. Actually, the linear chain is made of 2 CH_3 molecules at both extremes and $(p-2)$ CH_2 molecules in the inner chain

$$\begin{aligned} \mathrm{C} &= \mathrm{M}_1\text{–}\mathrm{M}_2\text{–}\cdots\text{–}\mathrm{M}_j\text{–}\cdots\text{–}\mathrm{M}_{p-1}\text{–}\mathrm{M}_p \\ &= \mathrm{CH}_3\text{–}\mathrm{CH}_2\text{–}\cdots\text{–}\mathrm{CH}_2\text{–}\cdots\text{–}\mathrm{CH}_2\text{–}\mathrm{CH}_3. \end{aligned} \tag{16}$$

It is remarkable to say that in the fuzzy-molecular inference model, the AHN is restricted to one univariate compound with one input $\mu_\Delta(x_1, \ldots, x_k)$ defined as (13) and one output y_i defined as (14). In case that a multiple-inputs-single-output (MISO) system has to be applied for a particular AHN, consider generalizing (1) as a multivariate function.

3.4. Knowledge Base. Since, the fuzzy-molecular inference model has a generic fuzzy inference engine, proper knowledge of a specific problem domain can be enclosed into the knowledge base (see Figure 2). For instance, this knowledge base is a matrix that summarizes all fuzzy rules of the form as in (11) in the following way.

(a) For all input variables $x_1, \ldots, x_k$, represent all possible combinations of them using the label of each set in the fuzzy partition of inputs, such that all antecedents in the fuzzy rules will be covered.

(b) For each combination (summary of antecedents), assign the corresponding label of molecule M_j that will act when the fuzzy rule is fired.

As an example of the knowledge base matrix construction, assume that there is a set of fuzzy rules like (17); thus, the knowledge base matrix for this particular system is shown in Table 1

$$\begin{aligned} R_1 &: \text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_2, \text{ then } y_1 \text{ is } \mathrm{M}_1, \\ R_2 &: \text{If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_1, \text{ then } y_2 \text{ is } \mathrm{M}_1, \\ R_3 &: \text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1, \text{ then } y_3 \text{ is } \mathrm{M}_2. \end{aligned} \tag{17}$$

3.5. Properties of the Fuzzy-Molecular Inference Model. The fuzzy-molecular inference model combines interesting properties from both fuzzy logic and artificial hydrocarbon networks. Advantages of the FMI-model are as the following.

(i) Fuzzy partitions in the output domain might be seen as linguistic units, for example, "low," "high."

(ii) Fuzzy partitions have a degree of understanding (parameters are metadata).

(iii) Molecular units deal with noise and uncertainties.

It is remarkable to say that molecules are excited by consequent values; thus, molecules do not model a given system, but transfer information from a fuzzy subspace to a crisp set. Moreover, molecular units have the property of filtering noise and uncertainties, especially important in real-world control applications, as described in Section 5.

Table 1: Knowledge base of (17).

x_1	x_2	y_i
A_1	B_2	M_1
A_2	B_1	M_1
A_1	B_1	M_2

In order to demonstrate the above advantages, an example of the application of the FIM-model in fuzzy control systems is provided in the following section. Then, Section 5 presents a case study that evaluates the performance of the FIM-model in a real application with dynamic uncertainties.

4. Design of Fuzzy-Molecular Based Controller for a DC Motor

In this section, the design of a velocity controller for a DC motor using the fuzzy-molecular inference model is described. The objective of this fuzzy-molecular controller is to show an example of how to apply the FMI-model as a fuzzy control system.

4.1. Definition of the DC Motor Model. For instance, consider a DC motor that regulates the velocity ω of its rotor varying the input voltage v. Let $G(s)$ be the transfer function of a given DC motor expressed in (18)

$$G(s) = \frac{1.5}{s^2 + 14s + 40.02}. \tag{18}$$

In order to simulate the performance of the DC motor, a discrete transfer function $G(z)$ was obtained using (18) and a sample time of 0.01 s. The discrete model of DC motor is shown in (19)

$$G(z) = \frac{7.16z + 6.83}{z^2 - 1.86z + 0.87} \times 10^{-5}. \tag{19}$$

Finally, if one supposes that DC motor is a causal, linear-time invariant system, then a difference equation of (19) can be expressed as (20), where ω is the velocity of the rotor, u is the input voltage, and k is the current sample time

$$\omega[k] = 1.86\omega[k-1] - 0.87\omega[k-2] + 7.16 \times 10^{-5} u[k] + 6.83 \times 10^{-5} u[k-1]. \tag{20}$$

4.2. Design of Control Law. The following control law is designed to achieve a step response of the DC motor model (20). Assuming the control diagram of Figure 3, the control law has two inputs—the error signal $\varepsilon(t)$ and the first derivative of error signal $\dot{\varepsilon}(t)$—and one output—the input voltage $u(t)$. Thus, a fuzzy-molecular PD controller will be designed.

Using the fuzzy-molecular inference model described in Section 3, the control law is formed by three blocks: *fuzzification*, *fuzzy inference engine,* and *defuzzification*, as follows.

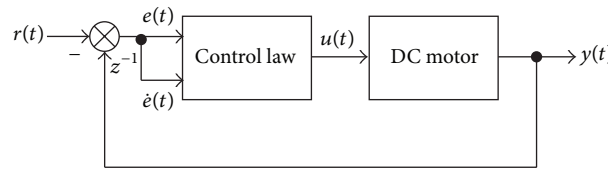


Figure 3: Block diagram of the PD control system implemented.

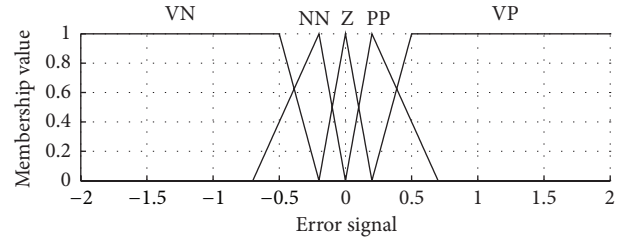


Figure 4: Fuzzy sets of the input error signal.

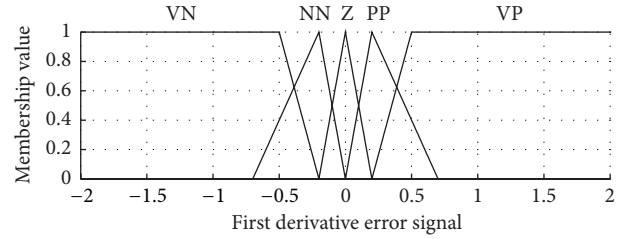


Figure 5: Fuzzy sets of the input first derivative error signal.

4.2.1. Fuzzification. The two input variables are partitioned into five fuzzy sets: "very negative" (VN), "negative" (NN), "zero" (Z), "positive" (PP), and "very positive" (VP). Figure 4 shows the fuzzy sets of the input variable $\varepsilon(t)$, and Figure 5 shows the fuzzy sets of the input variable $\dot{\varepsilon}(t)$. It is remarkable that parameters in the membership functions were tuned manually, and the input domain was previously normalized.

4.2.2. Fuzzy Inference Engine. The fuzzy inference engine for the fuzzy-molecular PD controller uses fuzzy rules of the form as in (11) with consequent values as in (14). In particular to the application, the implemented knowledge base is summarized in Table 2.

Notice that Table 2 reports, for each combination of input values, the fired molecule. For instance, the output signal was partitioned into five CH-molecules M_j, for all $j = 1, \ldots, 5$, that represent the action to be held. In particular, the output signal was partitioned into the following molecules: "very negative" (M_{VN}), "negative" (M_{NN}), "zero" (M_Z), "positive" (M_{PP}), and "very positive" (M_{VP}).

4.2.3. Defuzzification. The input voltage $u(t)$, the input signal of the plant, is the output variable that defines the last block of the fuzzy-molecular PD controller. In order to calculate the consequent values of fuzzy rules depicted in Section 4.2.2, the five CH-molecules are proposed in Figure 6 and were found using Algorithm 1. Notice that the output variable is

Table 2: Knowledge base of the fuzzy-molecular PD controller for DC motor.

ε	$\dot{\varepsilon}$				
	VN	NN	Z	PP	VP
VN	M_{VN}	M_{VN}	M_{VN}	M_{VN}	M_{VN}
NN	M_{VN}	M_{VN}	M_{VN}	M_{VN}	M_{NN}
Z	M_{VN}	M_{NN}	M_Z	M_{PP}	M_{VP}
PP	M_{PP}	M_{VP}	M_{VP}	M_{VP}	M_{VP}
VP	M_{VP}	M_{VP}	M_{VP}	M_{VP}	M_{VP}

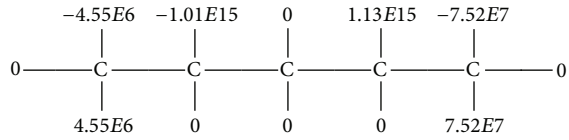


Figure 6: Artificial hydrocarbon network used in the fuzzy-molecular PD controller.

finally evaluated using (15), and resultant value is normalized. Finally, the adjustment parameters σ_j of CH-molecules are summarized in Table 3.

4.3. Results of the Velocity Control for a DC Motor. The fuzzy-molecular PD velocity controller for a DC motor described above was implemented and simulated. For instance, the objective of this application is to measure the performance of the system for a step response.

The system was subjected to a step function as shown in Figure 7. Results determine that the step response has 17% of maximum overshoot, a rise time of 0.19 s, a settling time of 0.45 s, and a maximum error of 0.002 in steady state. In order to measure the stability of the fuzzy-molecular PD controller, a phase diagram was obtained from the step response. Figure 8 shows the phase diagram of error signal versus derivative of error signal. As it can be seen in Figure 8, the fuzzy-molecular PD controller reaches a steady state near to the zero input state vector of the controller.

Then, the step was implemented with a reference signal varying in the range from −2 to 2 ×1000 rpm. After 10 seconds with a sample time of 0.01 s, the step response is depicted in Figure 9, where the light line represents the reference signal and the strong line represents the actual value of angular velocity of the rotor in the DC motor. As shown in Figure 9, the fuzzy-molecular PD controller has an excellent performance.

From the results obtained so far, it can be seen that the performance of the fuzzy-molecular PD controller has a very good quality (see Figure 8). The maximum overshoot, the settling time, and the maximum error in steady state correspond to the performance of a PD controller as reported in the literature of control theory [25].

On the other hand, the fuzzy-molecular PD controller was easily obtained. In this case, fuzzification was done via fuzzy sets tuned manually; however, there are other ways to find the optimal values of parameters in membership functions (see [7–9, 11, 23, 24]). In addition, defuzzification

Table 3: Adjustment parameters of molecules in the fuzzy-molecular PD controller.

CH-molecule	σ_j
M_{VN}	$+2.41E-14$
M_{NN}	$-4.97E-16$
M_Z	0.0
M_{PP}	$-4.43E-16$
M_{VP}	$-8.85E-17$

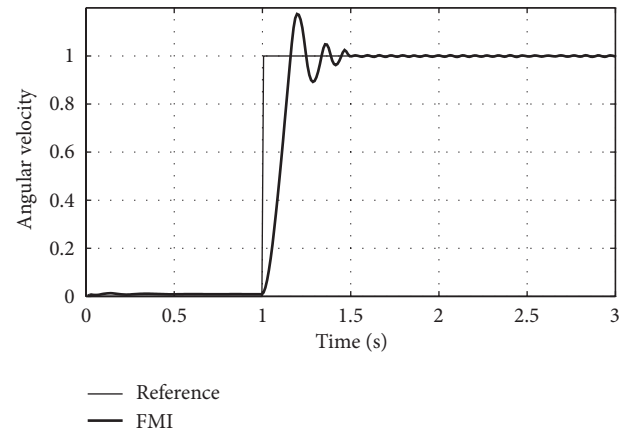


Figure 7: Step response of the fuzzy-molecular PD controller for DC motor.

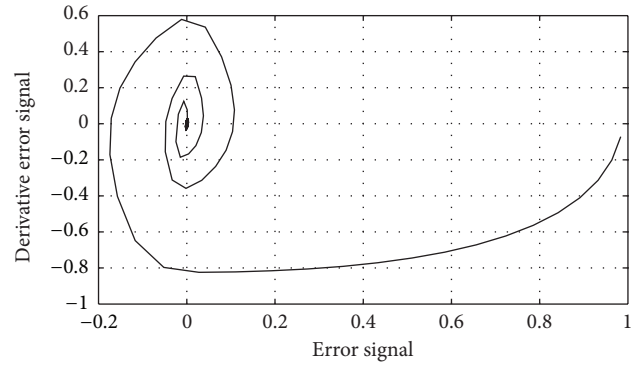


Figure 8: Phase diagram derivative error signal versus error signal in the step response of the fuzzy-molecular PD controller for DC motor.

was implemented with an artificial hydrocarbon network that depends on hydrogen and adjustment parameters that can be easily found using Algorithm 1.

5. Case Study: Fuzzy-Molecular Based Position Controller for a DC Motor

In this section, the design of a position controller for a DC motor using the fuzzy-molecular inference model is described. The objective of this case study is to improve type-2 fuzzy control systems using the fuzzy-molecular inference model.

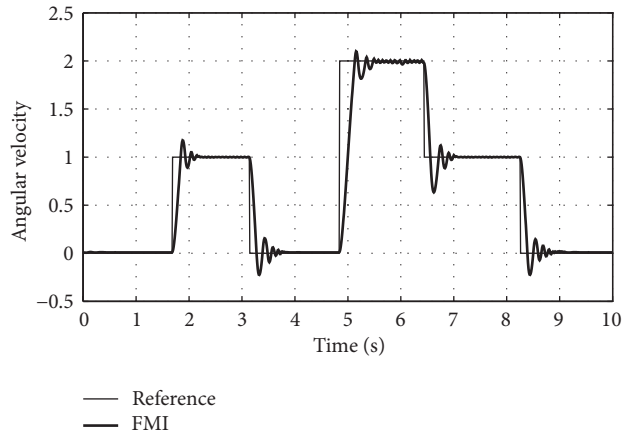


Figure 9: Step response of the fuzzy-molecular PD controller for DC motor.

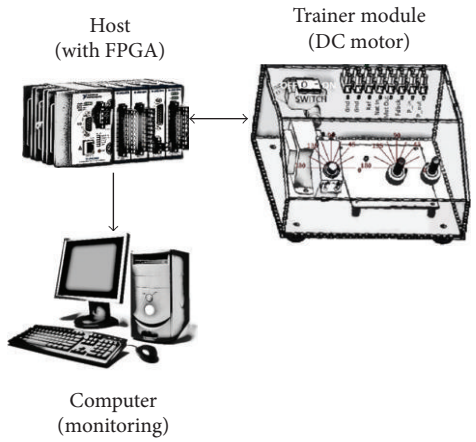


Figure 10: Overall system of the case study showing the trainer hardware module, the NI CompactRIO host, and the LabVIEW client for monitoring the system.

5.1. Description of the Hardware. The following case study was implemented on a trainer hardware module. It is prepared for sending reference signals (i.e., from a knob) and feedback signals (i.e., the current position of a DC motor) to a host in which a control law is running. The correction signal computed is sent back to the trainer module in order to feed a DC motor. In particular to this case study, a NI CompactRIO reconfigurable and embedded system based on field programmable gate arrays (FPGA) is used as the host. Figure 10 shows the overall system.

In addition, LabVIEW software is used for programming the control law on the NI CompactRIO and for monitoring the performance of the fuzzy-molecular control system.

On one hand, both the reference signal $r(t)$ that comes from a knob and the position signal $y(t)$ are in the voltage range $[0.0, 5.0]$ V, where 0.0 V represents an angle of 0° and 5.0 V represents an angle of 180°. On the other hand, the correction signal $u(t)$ is the input voltage of the DC motor in the range $[0.0, 5.0]$ V, where 0.0 V represents the maximum angular velocity of the motor to rotate counterclockwise,

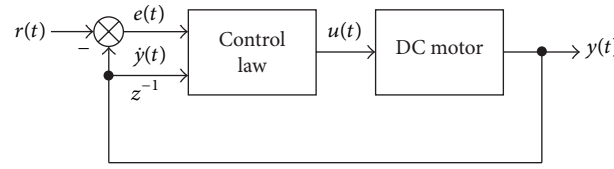


Figure 11: Block diagram of the position control system implemented.

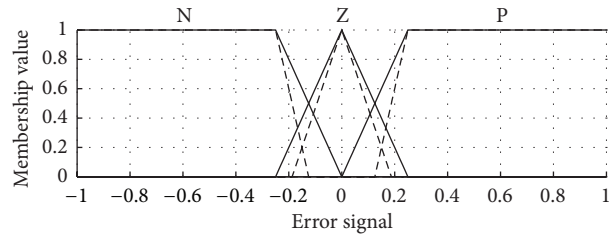


Figure 12: Fuzzy sets of the input error signal. Solid line: primary membership function. Dashed line: secondary membership function.

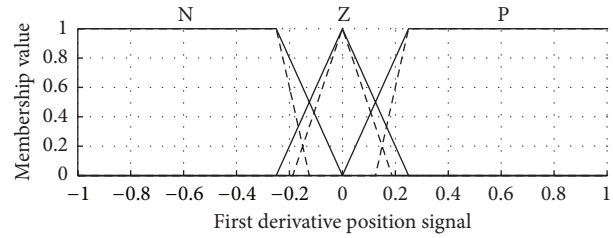


Figure 13: Fuzzy sets of the input first derivative position signal. Solid line: primary membership function. Dashed line: secondary membership function.

5.0 V represents the maximum angular velocity of the motor to rotate clockwise, and 2.5 V means no rotation.

It is remarkable to say that the position of the DC motor increases in counterclockwise direction and decreases in clockwise direction.

5.2. Design of Control Law. The following control law is designed to achieve a reference tracking response of the DC motor in the trainer model. Assuming the control diagram of Figure 11, the control law has two inputs—the error signal $\varepsilon(t)$ and the first derivative of the position signal $\dot{y}(t)$—and one output—the input voltage $u(t)$. Thus, a fuzzy-molecular PD controller will be designed.

Using the fuzzy-molecular inference model described in Section 3, the control law is designed as follows.

5.2.1. Fuzzification. The two input variables are partitioned into three type-2 fuzzy sets: "negative" (N), "zero" (Z), and "positive" (P). Figure 12 shows the fuzzy sets for input $\varepsilon(t)$, and Figure 13 shows the fuzzy sets for input $\dot{y}(t)$. It is remarkable to say that parameters in the membership functions were tuned manually.

Table 4: Knowledge base of the fuzzy-molecular position controller for the DC motor in the trainer module.

ε	$\dot{y}$		
	N	Z	P
N	M_{CW}	M_{CW}	M_{CW}
Z	M_{CW}	M_H	M_{CCW}
P	M_{CCW}	M_{CCW}	M_{CCW}

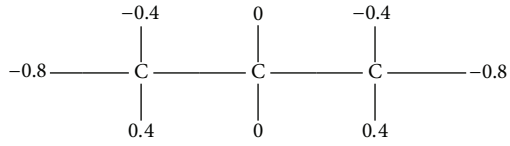


Figure 14: Artificial hydrocarbon network used in the fuzzy-molecular position controller.

As shown in Figures 12 and 13, type-2 fuzzy sets are determined by primary membership function $\mu_A^U(x)$ but also considers an additional value of uncertainty: the secondary membership function $\mu_A^L(x)$. The region inside these two membership functions is known as the footprint of uncertainty FOU [10, 14] as expressed in (21)

$$\text{FOU}(A) = \bigcup_{x \in X} \left(\mu_A^L(x), \mu_A^U(x) \right). \tag{21}$$

Then, two membership values (from primary and secondary functions) are computed for one input value. Moreover, if the secondary membership function coincides with the primary membership function, type-2 is reduced to an equivalent type-1 fuzzy system.

5.2.2. Fuzzy Inference Engine. The fuzzy inference engine for the fuzzy-molecular position controller uses fuzzy rules of the form as in (11) of both primary and secondary membership values $(\mu_A^L(x), \mu_A^U(x))$. Consequent values y_L and y_U are similarly obtained as (15) for both primary and secondary membership values, respectively. The resultant knowledge base is summarized in Table 4.

As noted in Table 4, the output signal was partitioned into three CH-molecules M_j, for all $j = 1, \ldots, 3$, that represent the action to be held. In particular, the output signal was partitioned into the following molecules: "clockwise" (M_{CW}), "halt" (M_H), and "counterclockwise" (M_{CCW}).

5.2.3. Defuzzification. In order to calculate the consequent values of fuzzy rules depicted in Table 4, the three CH-molecules are proposed in Figure 14 and were found using Algorithm 1. The adjustment parameters σ_j of CH-molecules are summarized in Table 5.

In this case study, the Nie-Tan method [26] is used for computing the final value of the output variable $u(t)$ for a type-2 fuzzy system because of its simplicity of computation. Other methods like Karnik-Mendel, Greenfield-Chiclana, or Wu-Mendel might be used [10, 12–14, 26]. The method

Table 5: Adjustment parameters of molecules in Figure 6.

CH-molecule	σ_j
M_{CW}	−1.0
M_H	0.0
M_{CCW}	1.0

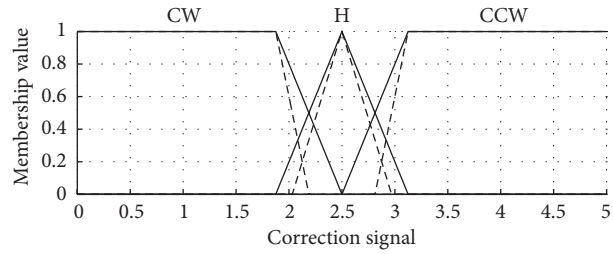


Figure 15: Fuzzy sets of the output correction signal. Solid line: primary membership function. Dashed line: secondary membership function.

generates a type reduction [26] of the form as in (22), where y is the crisp output value $u(t)$

$$y = \frac{y_L + y_U}{2}. \tag{22}$$

6. Results and Discussion

In order to demonstrate that the fuzzy-molecular inference model for fuzzy control systems can be used as an alternative of type-2 fuzzy control systems, two experiments were done. The first experiment considers a type-1 fuzzy controller system and the second experiment considers a type-2 fuzzy controller system. In both cases, the FIM-model based fuzzy control system designed in Section 5 is compared with a Mamdani's fuzzy controller system using the same parameters. The output variable was partitioned for the Mamdani's fuzzy controller system into three type-2 fuzzy sets: "clockwise" (CW), "halt" (H), and "counterclockwise" (CCW). Figure 15 shows this partition for the output variable $u(t)$.

6.1. Performance of the Type-1 Fuzzy-Molecular Controller. For this experiment, the fuzzy-molecular position controller for a DC motor described in Section 5 was reduced to a type-1 fuzzy system by only considering the primary membership functions in the fuzzification step, as well as in the Mamdani's fuzzy controller.

The system was subjected to a step function without noise as shown in Figure 16. Results of the FMI controller determine that it had a step response of 0% of maximum overshoot, a rise time of 1.0 s, and a maximum error of 2.5° in steady state. On the other hand, the system was subjected to a step function with 35% of noise as shown in Figure 17. Results of the FMI controller reports a 0% of maximum overshooting, a rise time of 1.1 s, and a maximum error of 5.8° in steady state measured from position 180°. For contrasting, Table 6

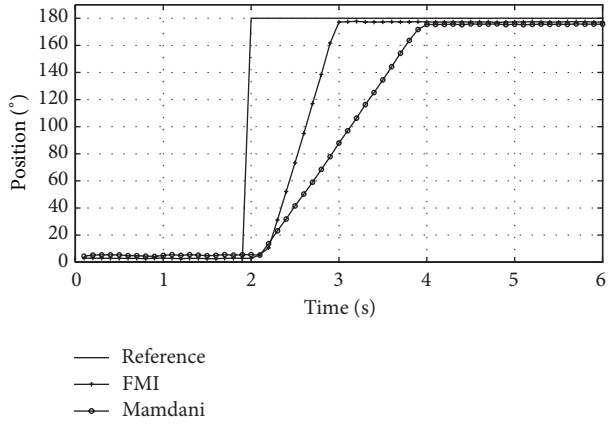


Figure 16: Step response without noise of FMI and Mamdani type-1 fuzzy controllers.

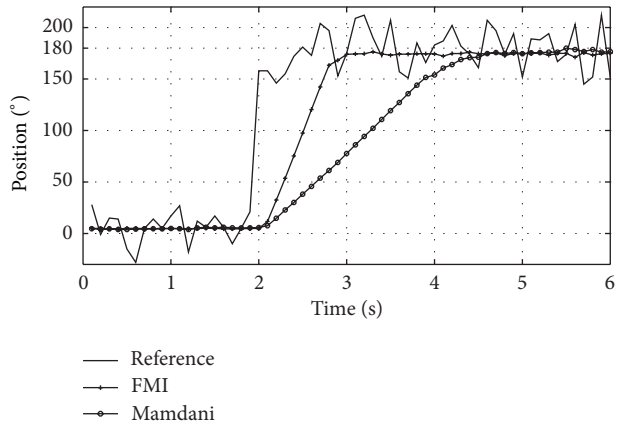


Figure 17: Step response with 35% noise of FMI and Mamdani type-1 fuzzy controllers.

summarizes the overall results of the FMI and Mamdani fuzzy controllers.

Notice in Figure 16 that the response of the FMI controller is 50% faster than the response of the Mamdani controller and has a less value of maximum error in steady state than then Mamdani controller. In comparison, Figure 17 shows that both fuzzy controllers are well stable as measured (5.8° and 5.5° of maximum error in steady state). However, FMI controller is still faster (1.1 s of rise time) than the response of the Mamdani controller (2.5 s of rise time). As noted, FMI controller has a better response for dynamic uncertainties than the Mamdani controller.

Also, the system was subjected to a ramp function without noise as shown in Figure 18. Results determine that the FIM controller has a maximum error of 3.6° in steady state while the Mamdani controller has 6.7°. On the other hand, the system was subjected to a ramp function with 35% of noise as shown in Figure 19. The FMI controller reports 11.0° of maximum error in steady state, and the Mamdani controller reports 12.3°. Also, Table 6 summarizes the overall results of this experiment with respect to the response of FMI and Mamdani fuzzy controllers.

Table 6: Experimental results of type-1 fuzzy controllers.

Fuzzy controller	Noise (%)	Rise time (s)	Steady-state error (°)
Step Response			
FIM	0.0	1.0	2.5
Mamdani	0.0	2.0	4.7
FIM	35.0	1.1	5.8
Mamdani	35.0	2.5	5.5
Ramp response			
FIM	0.0	—	3.6
Mamdani	0.0	—	6.7
FIM	35.0	—	11.0
Mamdani	35.0	—	12.3

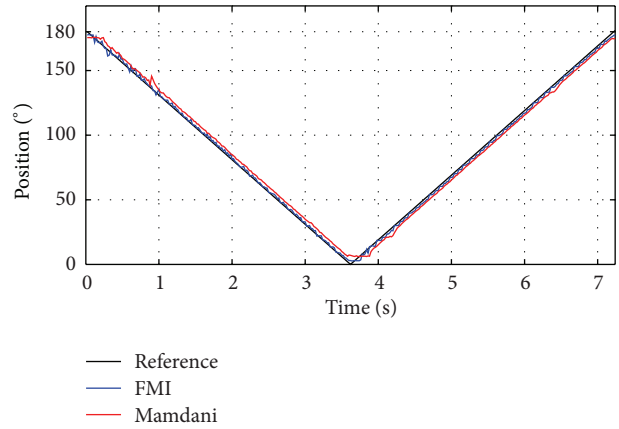


Figure 18: Ramp response without noise of FMI and Mamdani type-1 fuzzy controllers.

It is evident from Table 6 that both fuzzy controllers decrease their performance in presence of noise. However, the FIM controller can track the reference signal better than the Mamdani controller, as shown in the steady-state error. In addition, note that the FMI controller is slightly faster than the Mamdani controller.

6.2. Performance of the Type-2 Fuzzy-Molecular Controller. For this experiment, the type-2 fuzzy-molecular position controller for a DC motor described in Section 5 was implemented as well as the type-2 Mamdani controller.

Again, the system was subjected to a step function with 35% noise and without it as shown in Figures 20 and 21, respectively. The same process was done with a ramp function, and the responses of both controllers are shown in Figures 22 and 23, respectively. The overall results are summarized in Table 7.

As noted from Tables 6 and 7, the step responses of both FIM and Mamdani type-2 fuzzy controllers remain similar to type-1 controllers, as expected. Thus, type-1 and type-2 FIM fuzzy controllers are slightly equivalent with or without perturbations.

From Figures 22 and 23, it can be seen that the response of type-2 fuzzy controllers slightly better than type-1 controllers, as expected [10, 12–14, 26]. From the point of view of ramp

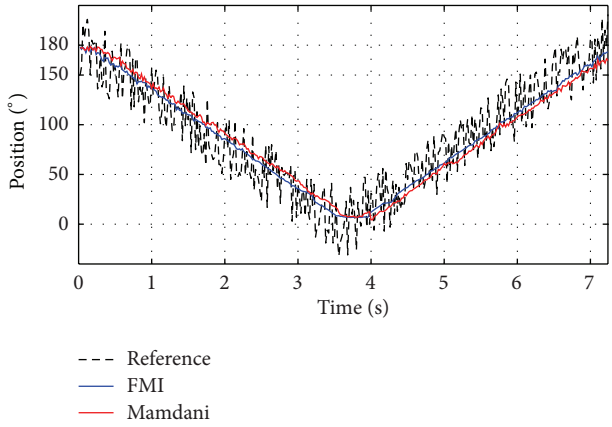


Figure 19: Ramp response with 35% noise of FMI and Mamdani type-1 fuzzy controllers.

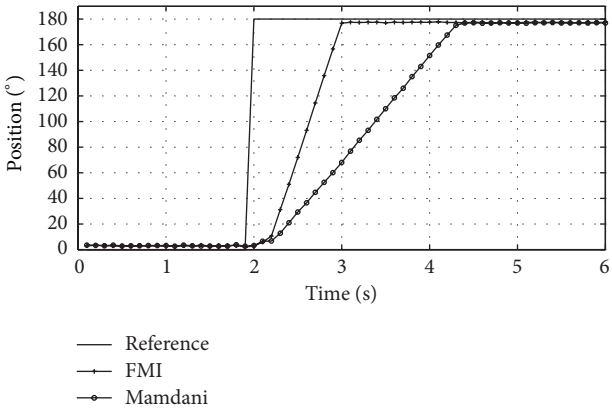


Figure 20: Step response without noise of FMI and Mamdani type-2 fuzzy controllers.

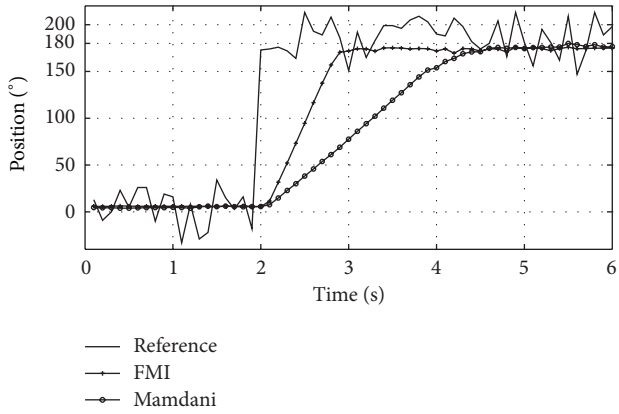


Figure 21: Step response with 35% noise of FMI and Mamdani type-2 fuzzy controllers.

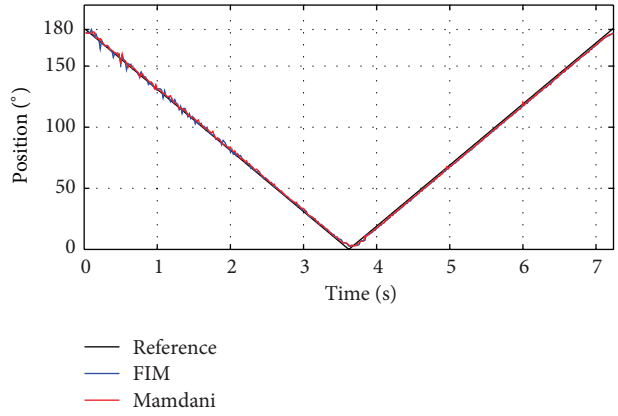


Figure 22: Ramp response without noise of FMI and Mamdani type-2 fuzzy controllers.

response, the FIM controller presents similar performance to the Mamdani controller without noise (3.8° and 3.7° maximum steady-state errors, resp.). Again, both controllers present the same tendency when they are exposed to noise, and in comparison with type-1 controllers, type-2 fuzzy controllers act slightly better as found in Tables 6 and 7 (FIM: 17.2% better, Mamdani: 1.7% better).

6.3. Discussion of FIM-Models. On one hand, from the above results, fuzzy-molecular inference models can achieve fuzzy control applications. Moreover, these FIM-model based controllers can be used as an alternative of type-2 fuzzy control systems. This statement comes from the evaluation and comparison of step and ramp responses between the FIM-controller designed in Section 5 and the Mamdani fuzzy controller; both models subjected to static and dynamic uncertainties. In this case study, a Mamdani's fuzzy control system was used because it is the fuzzy inference system most implemented in industry as reported in the literature [10, 14].

On the other hand, it is important to distinguish the fuzzy-molecular inference model from other fuzzy inference models like Takagi-Sugeno inference systems or Mamdani's fuzzy control systems [7–9, 11]. Thus, Table 8 presents a comparative chart of the FMI-model, Takagi-Sugeno's model, and Mamdani's model.

From Table 8, defuzzification process in each fuzzy inference model is different. As FMI-model uses artificial hydrocarbon networks, each molecule represents a linguistic partition of the output variable. In the above results, simple CH-molecules were implemented, but either complex molecules can be used. Thus, defuzzification can have complex nonlinear mappings in the FMI-model. In contrast, Takagi-Sugeno's model uses polynomial functions, and Mamdani's model represents linguistic partitions with membership functions associated with fuzzy sets. Parameters inside artificial hydrocarbon networks are hydrogen and adjustment values, polynomial coefficients for Takagi-Sugeno's model, and parameters of membership functions in Mamdani's model.

In addition, molecules in FMI-model make a mapping from membership or truth-values to output values also dealing with uncertainties. This is remarkable because Takagi-Sugeno's model maps from input values to output values,

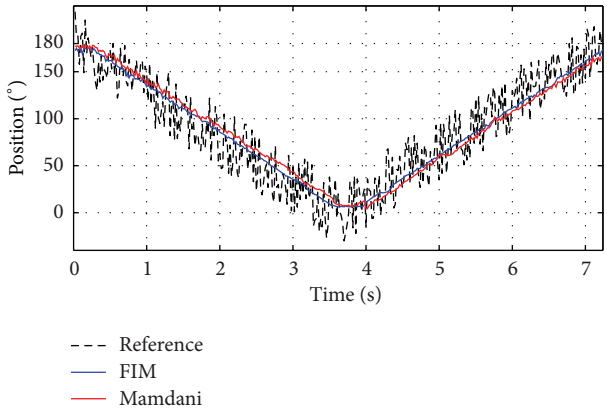


Figure 23: Ramp response with 35% noise of FMI and Mamdani type-2 fuzzy controllers.

Table 7: Experimental results of type-2 fuzzy controllers.

Fuzzy controller	Noise (%)	Rise time (s)	Steady-state error (°)
Step Response			
FIM	0.0	1.0	2.5
Mamdani	0.0	2.4	4.7
FIM	35.0	1.0	5.0
Mamdani	35.0	2.6	5.5
Ramp response			
FIM	0.0	—	3.8
Mamdani	0.0	—	3.7
FIM	35.0	—	9.1
Mamdani	35.0	—	12.1

Table 8: Comparative chart of different fuzzy inference models.

	FMI-model	Takagi-Sugeno	Mamdani
Defuzzification	AHNs	Polynomial functions	Membership functions
Definition of parameters	Hydrogen values Adjustment values	Coefficients	Parameters of membership functions
Mappings in defuzzification	Membership values to output values	Input values to output values	Membership values to output values

and using fuzzy inference values linearly acts on the final output value. At last, Mamdani's model makes a mapping from membership values to output values. In fact, the fuzzy-molecular inference model combines linguistic partitions of output variables with molecular structures.

7. Conclusions

In this paper, a new fuzzy algorithm based on artificial hydrocarbon networks called fuzzy-molecular inference model (FMI-model) was proposed, taking advantage of the power of molecular units of information. In this approach, molecules of artificial hydrocarbon networks are implemented as fuzzy partitions in the output domain. Since the FMI-model is based on AHNs, properties of molecules are inherited. Two characteristics of the proposed fuzzy-molecular inference model are both interpretation of linguistic molecules and partial understanding of fuzzy partitions via metaparameters in AHNs.

In that way, the novel fuzzy algorithm treats molecules as fuzzy partitions of the output variable, transferring information from a fuzzy subspace to a crisp set, allowing to set the number of fuzzy partitions linguistically, but also these molecules are characterized by hydrogen values that can be referred to as meta-data information, giving the opportunity to partially understand the molecular behavior. Moreover, the proposed fuzzy-molecular inference model has some advantages in comparison with other fuzzy inference systems. For instance, FMI-model occupies parameters with meta-data information in comparison with Mamdani's inference system in which parameters associated with membership functions do not reveal important information of the fuzzy partition. If parameters are meta-data information, it is easier to tune fuzzy partitions because both experts and real data information coming from the system can be combined into a single unit, no matter how complex the mapping is. In addition, since FMI-model does not model a given system like Takagi-Sugeno's inference system, it preserves a more natural way of defuzzification from a fuzzy subspace to a crisp set. Finally, since molecules in artificial hydrocarbon networks can filter information [18, 19], the fuzzy-molecular inference model also shares this property allowing to deal with uncertain data.

Thus, the proposed fuzzy-molecular inference model has three steps: fuzzification, fuzzy inference engine, and defuzzification. Specially, molecules are mappings from implication values to output variables. In addition, in this work, a linear chain of CH-primitive molecules was used, but the FMI-model allows complex molecules associated with each fuzzy rule handling complex nonlinear mappings from fuzzy subspaces to crisp sets.

On the other hand, the proposed model was applied to control the angular velocity of a simulated DC motor in which the results confirm that the FMI-model can be used in control applications. Furthermore, a case study was presented in which the FMI-model was used for controlling the position of a real DC motor. Experimental results demonstrate that fuzzy-molecular based control systems can deal with uncertainties as type-2 fuzzy control systems do. Then, it suggests that FMI-based controllers can be used as an alternative of type-2 fuzzy control systems. In practical applications where hardware restricts the operational computations or memory storage, FMI-based controllers can be implemented because of its simplicity.

Future research considers the design of training procedures for optimality in molecules at the defuzzification stage of FMI-models. In addition, since artificial hydrocarbon networks are considered under the class of learning algorithms, the usage of molecular units in FMI-models might be applied for online adaptation (learning and evolution) of the overall fuzzy control system to improve its performance.

Acknowledgments

This work was supported by a scholarship award from Tecnológico de Monterrey, Campus Ciudad de México and a scholarship for living expenses from CONACYT.

References

[1] L. Xia and Y. Xiuju, "The application of adaptive fuzzy inference model in the nonlinear dynamic system identificatio," in *Proceedings of the 2nd International Conference on Intelligent Computing Technology and Automation (ICICTA '09)*, pp. 814–817, October 2009.

[2] H. Seki, "Type-2 fuzzy functional SIRMs connected inference model," in *Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems and 13th International Symposium on Advanced Intelligent Systems*, pp. 1615–11620, 2012.

[3] K.-E. Ko and K.-B. Sim, "An EEG signals classification system using optimized adaptive neuro-fuzzy inference model based on harmony search algorithm," in *Proceedings of the 11th International Conference on Control, Automation and Systems (ICCAS '11)*, pp. 1457–1461, October 2011.

[4] Y.-P. Huang, T.-W. Chang, and F.-E. Sandnes, "Improving image retrieval efficiency using a fuzzy inference model and genetic algorithm," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 361–366, June 2005.

[5] C. Gong and C. Han, "Robust H_∞ control of uncertain T-S fuzzy time-delay system: a delay decomposition approach," *Mathematical Problems in Engineering*, vol. 2013, Article ID 345601, 10 pages, 2013.

[6] W. Huang and S.-K. Oh, "Identification of fuzzy inference systems by means of a multiobjective opposition-based space search algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 725017, 13 pages, 2013.

[7] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[8] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *IEEE Proceedings of the Institution of Electrical Engineers*, vol. 121, no. 12, pp. 1585–1588, 1974.

[9] Y. Tsukamoto, "An approach to fuzzy reasoning method," in *Advances in Fuzzy Set Theory and Applications*, M. Gupta, R. Ragade, and R. Yager, Eds., pp. 137–149, North-Holland, Amsterdam, The Netherlands, 1979.

[10] I. Iancu, "A mamdani type fuzzy logic controller," in *Fuzzy Logic: Controls, Concepts, Theories and Applications*, pp. 325–350, InTech, 2012.

[11] J.-S. R. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.

[12] O. Linda and M. Manic, "Comparative analysis of Type-1 and Type-2 fuzzy control in context of learning behaviors for mobile robotics," in *Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society (IECON '10)*, pp. 1092–1098, November 2010.

[13] S. Musikasuwan and J. M. Garibaldi, "On relationships between primary membership functions and output uncertainties in interval type-2 and non-stationary fuzzy sets," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1433–1440, July 2006.

[14] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.

[15] H. Zhang and D. Liu, *Fuzzy Modeling and Fuzzy Control*, Springer, Boston, Mass, USA, 2006.

[16] H. Ponce and P. Ponce, "Artificial organic networks," in *Proceedings of the IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA '11)*, pp. 29–34, November 2011.

[17] H. Ponce and P. Ponce, "Artificial hydrocarbon networks," in *Proceedings of the 9th International Conference on Innovation and Technological Development (CIINDET '11)*, pp. 614–618, 2011.

[18] H. Ponce, P. Ponce, and A. Molina, "A novel adaptive filtering for audio signals using artificial hydrocarbon networks," in *Proceedings of the 9th International Conference on Electrical Engineering, Computing Science and Automation Control*, pp. 277–282, 2012.

[19] H. Ponce and P. Ponce, "Artificial hydrocarbon networks: a new algorithm bio-inspired on organic chemistry," *International Journal of Artificial Intelligence and Computational Research*, vol. 4, no. 1, pp. 39–51, 2012.

[20] H. Ponce, P. Ponce, and A. Molina, "A new training algorithm for artificial hydrocarbon networks using an energy model of covalent bonds," in *Proceedings of the IFAC Conference on Manufacturing Modelling, Management and Control*, 2013.

[21] H. Ponce, P. Ponce, and A. Molina, "Artificial hydrocarbon networks: a bio-inspired computational algorithm for modeling problems," Tech. Rep., Graduate School of Engineering, Tecnológico de Monterrey, Campus Ciudad de México, Mexico City, Mexico, 2013.

[22] H. Ponce and P. Ponce, "A Novel Approach of Artificial Hydrocarbon Networks on Adaptive Noise Filtering for Audio Signals," Tech. Rep., Graduate School of Engineering, Tecnológico de Monterrey, Campus Ciudad de México, Mexico City, Mexico, 2013.

[23] L. M. de Campos and S. Moral, "Learning rules for a fuzzy inference model," *Fuzzy Sets and Systems*, vol. 59, no. 3, pp. 247–257, 1993.

[24] Y.-P. Huang, S.-H. Yu, and M.-S. Horng, "An efficient tuning method for designing a fuzzy inference model," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1–6, October 2001.

[25] K. Ogata, *Modern Control Engineering*, Prentice Hall, Englewood Cliffs, NJ, USA, 2010.

[26] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1425–1432, June 2008.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 956978, 14 pages
http://dx.doi.org/10.1155/2013/956978

Research Article

A Fuzzy-Neural Ensemble and Geometric Rule Fusion Approach for Scheduling a Wafer Fabrication Factory

Hsin-Chieh Wu[1] and Toly Chen[2]

[1] *Department of Industrial Engineering and Management, Chaoyang University of Science and Technology, Taiwan*
[2] *Department of Industrial Engineering and Systems Management, Feng Chia University, Taiwan*

Correspondence should be addressed to Toly Chen; tcchen@fcu.edu.tw

Received 22 March 2013; Accepted 13 June 2013

Academic Editor: Pedro Ponce

In this study, the fuzzy-neural ensemble and geometric rule fusion approach is presented to optimize the performance of job dispatching in a wafer fabrication factory with an intelligent rule. The proposed methodology is a modification of a previous study by fusing two dispatching rules and diversifying the job slacks in novel ways. To this end, the geometric mean of the neighboring distances of slacks is maximized. In addition, the fuzzy c-means (FCM) and backpropagation network (BPN) ensemble approach was also proposed to estimate the remaining cycle time of a job, which is an important input to the new rule. A new aggregation mechanism was also designed to enhance the robustness of the FCM-BPN ensemble approach. To validate the effectiveness of the proposed methodology, some experiments have been conducted. The experimental results did support the effectiveness of the proposed methodology.

1. Introduction

The scheduling of complex manufacturing systems is usually a NP-hard problem (see Table 1), which means it is very difficult for the production controller to find the best schedule within a reasonable period of time. To resolve this problem, a viable strategy is to develop a digital manufacturing model for the complex manufacturing system, based on which production simulation can be carried out to search for the optimal schedule in an efficient way. Based on this viewpoint, this study attempts to optimize the performance of job dispatching in a wafer fabrication factory, with the aid of the digital manufacturing model of the wafer fabrication factory. In this field, a digital manufacturing model is mainly used for three purposes—to evaluate the performance of a scheduling method, to optimize the scheduling performance, and to generate some test data for the subsequent optimization.

A wafer fabrication factory is a very complex manufacturing system featured by thousands of machines, tens of product types, various priorities, reentrant process flows, uncertain demand, and others. In addition, building a wafer fabrication factory usually needs billions of dollars. The capital-intensive feature makes the efficient use of a wafer fabrication factory through good production management a very important task. However, some studies [1–3] noted that job scheduling in a semiconductor manufacturing factory is a very difficult task. As a result, many wafer fabrication factories suffer from lengthy cycle times. It is therefore not possible to promise their customers an attractable due date. On the other hand, owing to the uncertainty of the cycle time, the risk of missing the due date is also high.

Many different methods can be used for job dispatching in a wafer fabrication factory, such as dispatching rules, heuristics, data mining-based approaches [4, 5], agent technologies [4, 6–8], and simulation. Gupta and Sivakumar [9] classified the approaches for scheduling a wafer fabrication system into four categories: heuristic rules, mathematical, programming techniques, neighborhood search methods, and artificial intelligence techniques. A lot of studies have committed to the development of advanced dispatching rules for a wafer fabrication factory. For example, Altendorfer et al. [10] proposed the work in parallel queue (WIPQ) rule to maximize the throughput at a low level of work in process (WIP). Zhang et al. [11] proposed the dynamic bottleneck detection (DBD) approach, in which workstations are classified into

Table 1: The complexity of a decision problem.

Complexity	Meaning
P	A decision problem that is polynomial solvable.
NP	A decision problem that can be verified by a polynomial algorithm if a proper clue is given.
NP-hard	A decision problem P is called NP-hard if the entire class NP polynomially reduces to problem P.
NP-complete	A decision problem P is called NP-complete if it is both in classes NP and NP-hard.

several categories and then different dispatching rules, including first-in-first out-(FIFO), the shortest processing time until the next bottleneck (SPNB), and critical ratio (CR), are applied to these categories. Cao et al. [12] proposed a drum-buffer-rope based scheduling method for semiconductor manufacturing systems, which was focused on the control of the bottleneck machines. According to Lee et al. [13], the past experiences of scheduling were emphasized, and Petri nets were used to model the semiconductor manufacturing activities. As a result, the scheduling of a semiconductor manufacturing process was based on the token movements in the corresponding Petri net. Wang et al. [14] considered the scheduling of unrelated parallel machines in semiconductor manufacturing. After the problem reduction, some local heuristics could be proposed. Chen [15] smoothed the fluctuation in the estimated remaining cycle time, balanced it with that of the mean release rate, and proposed the nonlinear fluctuation smoothing policy for mean cycle time (NFSMCT). Hu et al. [16] divided the process flow into several stages and protected the bottleneck step at each reentrant stage from the system fluctuations. Although these dispatching rules are relatively easier to use, they cannot produce optimal or near-optimal scheduling results. To this end, a scheduling problem has to be formulated as a mathematical programming problem. The optimal solution of the mathematical programming problem gives the optimal scheduling of the manufacturing system. However, the mathematical programming problem of scheduling a wafer fabrication factory is too large and can hardly be solved. To resolve this problem, several treatments have been taken in the literature.

The first way is to choose the most suitable dispatching rule from the existing rules for the wafer fabrication factory dynamically. Chern and Liu [17] proved that general family-based scheduling rules are better than the individual job scheduling rule in terms of machine utilization for multi-server and multiple-job reentrant production systems under some conditions. Hsieh et al. [5] chose one rule from the fluctuation smoothing policy for mean cycle time (FSMCT), the fluctuation smoothing policy for cycle time variation (FSVCT), largest deviation first (LDF), one step ahead (OSA), and FIFO. The selection was based on the results of extensive production simulation.

The second way is to add adjustable parameters that can be optimized to the dispatching rule. For example, Chen [18] proposed the one-factor tailored NFSMCT (1f-TNFSMCT) rule and the one-factor tailored nonlinear FSVCT (1f-TNFSVCT) rule. Various values of the parameter were tried in production simulation, and the one giving the best performance was chosen. Chen [19] used more parameters and proposed 2f-TNFSMCT and 2f-TNFSVCT. Both Dabbas and Fowler [20] and Dabbas et al. [21] combined some dispatching rules into a single rule by forming their linear combination with relative weights. However, a systematic procedure to determine the weights of these rules was missing. Similarly, Chiang et al. [22] developed a genetic algorithm (GA) for generating good dispatching rules through combining multiple rules linearly. Chen and Wang [23] proposed a biobjective nonlinear fluctuation smoothing rule with an adjustable factor (1f-biNFS) to optimize both the average cycle time and cycle time standard deviation at the same time. More degrees of freedom seem to be helpful for the performance of customizable rules. For this reason, Chen et al. [24] extended 1f-biNFS to a biobjective fluctuation smoothing rule with four adjustable factors (4f-biNFS). One drawback of these methods is that only static factors are used, and they must be determined in advance. To this end, most studies (e.g., [15–20]) performed extensive simulations. This is not only time consuming but also fails to consider enough possible combinations of these factors. To solve this problem, Kim et al. [25] suggested three techniques for accelerating rule comparison using production simulation.

The third way is to estimate the best schedule from a limited simulation results. In this manner, only a few combinations of the adjustable parameters will be tried in the production simulation, and then some estimation technique is applied to estimate the scheduling performance from the parameter values. To this end, Dabbas et al. [26] applied the design of experiment (DOE) techniques as well as the desirability function approach. In Li and Sigurdur [27], historic schedules were transformed into appropriate data files that were mined in order to find out which past scheduling decisions corresponded to the best practices. Harrath et al. [7] proposed a hybrid genetic algorithm (GA) and data mining approach to determine the optimal scheduling plan of a job shop, in which GA was used to generate a learning population of good solutions. These good solutions were then mined to find out some decision rules that could be transformed into a metaheuristic. Koonce and Tsai [4] proposed a similar methodology. Chen [28] attempted to relate the scheduling performance to the parameter values using a backpropagation network (BPN). However, the explanatory ability of the BPN was not as good as expected. Recently, Chen [29] constructed a nonlinear programming (NLP) model to optimize the parameter values in 2f-TNFSMCT and 2f-TNFSVCT by maximizing the standard deviation of slacks, which was considered to reduce possible ties. However, the NLP model was very difficult to optimize and required a lot of time to solve.

In short, the existing approaches have the following problems.

(1) A more effective approach to optimize the parameter values is needed.

Table 2: The differences between the proposed methodology and the previous methods.

Method	Way of fusing rules	Estimating the remaining cycle time	Optimization method	Heuristic	Digital manufacturing model is used for	Required evaluation rime
Hsieh et al. [5]	NA	NA	NA	Simulation	Evaluating/comparing scheduling performances	Lengthy
Dabbas et al. [26]	Linear	NA	DOE + the desirability function	Simulation	Evaluating/comparing scheduling performances	Lengthy
Harrath et al. [7]	Linear	NA	Enumeration	Association rule	Evaluating/comparing scheduling performances	Lengthy
Chen [29]	Nonlinear	FCM-BPN	Maximize the standard deviation of slacks	NLP	Confirmation study	Moderate
The proposed methodology	Nonlinear	FCM-BPN ensemble	Maximizing the geometric mean of the neighboring distances of slacks	NLP + systematic procedure	Confirmation study	Short

(2) Maximizing the standard deviation of slacks may lead to the situation that most slacks concentrate on the two extremes.

(3) How to avoid carrying out large-scale, time-consuming production simulation experiment is worth exploring.

(4) New applications of the digital manufacturing model are still to be discovered.

In order to solve some of these problems and to further improve the performance of job scheduling in a wafer fabrication factory, Chen's approach has been modified, and a fuzzy-neural-ensemble and geometric rule fusion approach was proposed in this study. The proposed methodology has the following new characteristics.

(1) To diversify the slack, the geometric mean of the neighboring distances of slacks is to be considered, rather than the standard deviation.

(2) Diversifying the slacks leads to a NLP problem, which is not easy to solve. To solve this problem, a systematic procedure has been established to search the optimal solution of the NLP problem in an effective manner.

(3) The fuzzy c-means (FCM) and backpropagation network (BPN) ensemble approach [30] was also modified to estimate the remaining cycle time of a job. With a novel aggregation mechanism, the FCM-BPN ensemble approach is not only accurate but also robust to the untrained data.

(4) The two existing rules, 2f-TFSMCT and 2f-TNFSVCT, are fused in a novel way so that two objectives, the average cycle time and cycle time variation, can be improved at the same time. Data fusion is a technique that gathers information and combines data from multiple sources in order to achieve inferences. This method is more efficient and potentially more accurate than if the inference was achieved from a single source.

The differences between the proposed methodology and the previous methods are summarized in Table 2.

The outline of this paper is described as follows. First, in Section 2, a new rule is proposed through the fusion of the two rules, 2f-TNFSMCT and 2f-TNFSVCT, in a novel way. To determine the values of parameters in the new rule, a NLP model is established. A systematic procedure is then used to solve the NLP problem. In addition, the remaining cycle time of each job is a necessary input of the new rule. The FCM and BPN ensemble approach, therefore, is applied to estimate the remaining cycle time of a job. In Section 3, the performance of the proposed methodology is evaluated by carrying out some experiments. Finally, this paper is concluded in Section 4.

2. Methodology

The variables and parameters that will be used in this study are defined in the Abbreviations section.

The proposed methodology includes the following six steps.

Step 1. Fuse the two rules, 2f-FSMCT and 2f-FSVCT, to form the new rule 2f-biFS.

Step 2. Establish a NLP model to optimize the parameters of the new rule.

Step 3. Apply a systematic procedure to solve the NLP problem.

Step 4. Estimate the remaining cycle time of a job using the FCM-BPN ensemble approach.

Step 5. Incorporate the estimated remaining cycle time into the new rule.

Step 6. Carry out experiments to evaluate the performance of the new rule.

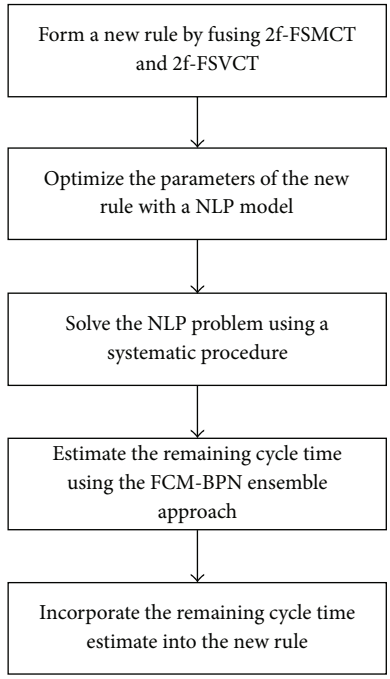


Figure 1: The flow chart of the proposed methodology.

The flow chart of the proposed methodology is shown in Figure 1. The algorithm proposed in this study is therefore indicated with FCM-BPN-ensemble-2f-biFS. Some steps correspond to the operations of the digital manufacturing model, as summarized in Table 3.

2.1. The Wafer Fabrication Environment. In this study, a wafer fabrication factory located in Taichung Scientific Park of Taiwan is analyzed. The wafer fabrication factory has a monthly capacity of about 25,000 wafers. Currently, more than 10 types of memory products are being produced with 58 nm~110 nm technologies in the wafer fabrication factory. For this, more than 500 workstations are used. Machine failure and repairing times basically follow the exponential and uniform distributions. Jobs released into the fabrication factory are assigned three types of priorities, that is, "normal", "hot" and "super hot". Jobs with the highest priorities will be processed first. The current scheduling policy in the wafer fabrication factory is FIFO. As a result, the longest average cycle time exceeds three months with a variation of more than 300 hours. The wafer fabrication factory is therefore seeking better scheduling rules to replace FIFO.

2.2. The New Rule. Information fusion has been widely used in scheduling and other system control purposes. According to Zhao et al. [31], the historical data of passenger flow of urban rail transportation were gathered and fused, in order to calculate the characteristic data for real-time schedule. Lebret et al. [32] presented an intelligence fusion system based on gain scheduling control and hybrid neural networks for system fault detection and fault-tolerant control. Li et al. [33] measured and fused the collected raw data for the scheduling of a multihop hierarchical sensor network. In this study, the two basic fluctuation smoothing (FS) rules [34] are to be fused, in order to have a better control over the scheduling of the wafer fabrication factory. The first one is FSMCT, in which the slack of a job is defined as

Table 3: Operations of the digital manufacturing model in the proposed methodology.

Step	Operation of the digital manufacturing model
The new rule	Control
FCM	Reduction
BPN	Approximation
The aggregation mechanism	Combining submodels

$$\mathrm{SKM}_{ju} = \frac{j}{\lambda} - \mathrm{RCTE}_{ju} \tag{1}$$

which is aimed at the reduction of the average cycle time. Reducing the cycle time is a critical task in managing a wafer fabrication factory, especially, in a mass production setting such as in a memory fabrication factory [35]. The second rule is FSVCT:

$$\mathrm{SKV}_{ju} = R_j - \mathrm{RCTE}_{ju} \tag{2}$$

which is aimed at the minimization of cycle time standard deviation. Some variants of the two rules were summarized in Table 4 [18, 19, 23, 36, 37]. Jobs with smaller slack values (SKM_{ju} or SKV_{ju}) will be processed earlier. However, a tie is formed if more than one job has the same slack, which results in scheduling difficulties.

Chen [19] normalized the parameters in the FS rules and then divided them, which led to the 2f-TNFSMCT rule:

$$\begin{aligned}\mathrm{SKM}_{ju} = {} & \left(\frac{\lambda\beta}{(n-1)\left(\mathrm{RCTE}_{ju} - \min\left(\mathrm{RCTE}_{ju}\right)\right)}\right)^{\xi} \\ & \cdot \left(\frac{j}{\lambda} - \mathrm{RCTE}_{ju} + \zeta\left(\mathrm{RCTE}_{ju} - \frac{1}{\lambda}\right)\right)\end{aligned} \tag{3}$$

and the 2f-TNFSVCT rule:

$$\begin{aligned}\mathrm{SKV}_{ju} = {} & \left(\frac{\beta}{\alpha\left(\mathrm{RCTE}_{ju} - \min\left(\mathrm{RCTE}_{ju}\right)\right)}\right)^{\xi} \\ & \cdot \left(R_j - \mathrm{RCTE}_{ju} + \zeta\left(\mathrm{RCTE}_{ju} - \min\left(R_j\right)\right)\right),\end{aligned} \tag{4}$$

where

$$\begin{aligned}\alpha &= \max\left(R_j\right) - \min\left(R_j\right) \\ \beta &= \max\left(\mathrm{RCTE}_{ju}\right) - \min\left(\mathrm{RCTE}_{ju}\right)\end{aligned} \tag{5}$$

$0 \le \xi, \zeta \le 1$. In Chen's study, three models were established to form the combination of ξ and ζ. For example,

$$(\text{linear model})\ \xi = \zeta, \tag{6}$$

$$(\text{nonlinear model})\ \xi = \zeta^k, \quad k \ge 0, \tag{7}$$

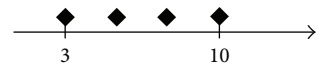
$$(\text{logarithmic model})\ \xi = \frac{\ln(1+\zeta)}{\ln 2}. \tag{8}$$

Table 4: Some variants of FSMCT and FSVCT.

Rule	
NFS [23]	$\mathrm{SK}_{ju} = \dfrac{\left(\left(R_j - \min\left(R_j\right)\right) / \left(\max\left(R_j\right) - \min\left(R_j\right)\right)\right)^{\xi} \cdot \left((j-1)/(n-1)\right)^{1-\xi}}{\left(\mathrm{RCTE}_{ju} - \min\left(\mathrm{RCTE}_{ju}\right)\right) / \left(\max\left(\mathrm{RCTE}_{ju}\right) - \min\left(\mathrm{RCTE}_{ju}\right)\right)}$
1f-TNFSMCT [18]	$\mathrm{SKV}_{ju} = \left(\dfrac{\lambda\beta}{(n-1)\left(\mathrm{RCTE}_{\mathrm{ju}} - \min\left(\mathrm{RCTE}_{ju}\right)\right)}\right)^{\xi} \cdot \left(\dfrac{j}{\lambda} - \mathrm{RCTE}_{ju} + \xi\left(\mathrm{RCTE}_{ju} - \dfrac{1}{\lambda}\right)\right)$
1f-TNFSVCT [18]	$\mathrm{SKM}_{ju} = \left(\dfrac{\beta}{\alpha\left(\mathrm{RCTE}_{\mathrm{ju}} - \min\left(\mathrm{RCTE}_{\mathrm{ju}}\right)\right)}\right)^{\xi} \cdot \left(R_j - \mathrm{RCTE}_{\mathrm{ju}} + \xi\left(\mathrm{RCTE}_{\mathrm{ju}} - \min\left(R_j\right)\right)\right)$
2f-TNFSMCT [19]	$\mathrm{SKV}_{ju} = \left(\dfrac{\lambda\beta}{(n-1)\left(\mathrm{RCTE}_{\mathrm{ju}} - \min\left(\mathrm{RCTE}_{ju}\right)\right)}\right)^{\xi} \cdot \left(\dfrac{j}{\lambda} - \mathrm{RCTE}_{ju} + \zeta\left(\mathrm{RCTE}_{ju} - \dfrac{1}{\lambda}\right)\right)$
2f-TNFSVCT [19]	$\mathrm{SKM}_{ju} = \left(\dfrac{\beta}{\alpha\left(\mathrm{RCTE}_{\mathrm{ju}} - \min\left(\mathrm{RCTE}_{\mathrm{ju}}\right)\right)}\right)^{\xi} \cdot \left(R_j - \mathrm{RCTE}_{\mathrm{ju}} + \zeta\left(\mathrm{RCTE}_{\mathrm{ju}} - \min\left(R_j\right)\right)\right)$

Standard deviation maximized

Figure 2: The results of maximizing the standard deviation of slacks.



The geometric mean of the neighboring distances maximized

Figure 3: The results of maximizing the geometric mean of the neighboring distances of slacks.

However, no model can guarantee the optimization of the scheduling performance. To solve this problem, Wang et al. [37] advocated that, through diversifying the slacks of jobs, the overlapping of slacks can be reduced, which avoids mis-scheduling and is conducive to the scheduling performance. To this end, Wang maximized the standard deviation of slacks. However, this treatment may lead to the situation that most slacks concentrate on the two extremes [29], as illustrated in Figure 2.

In this study, to diversify the slack, the geometric mean of the neighboring distances of slacks is maximized instead:

$$\text{Max } Z = \sqrt[n-1]{\prod_{j=2}^{n}\left(\mathrm{SK}_{(j)u} - \mathrm{SK}_{(j-1)u}\right)}. \tag{9}$$

An example is illustrated in Figure 3. Obviously, to maximize the geometric mean of the neighboring distances, the slacks need to be evenly distributed.

However, (9) is not easy to solve. To tackle this, the following procedure is established.

(1) Change the values of parameters.

(2) Derive the slack of each job.

(3) Exclude jobs with very large or small slacks.

(4) Sort the slacks of the remaining jobs.

(5) Calculate the distance between each neighboring pair.

(6) Add a small value Δ, for example, 0.01, to the distance to avoid the case of zero slack.

(7) Obtain the geometric mean of the distances.

(8) If the geometric mean has been maximized, stop; otherwise, return to Step 1.

As a result, in 2f-TNFSMCT, the following objective function is to be optimized:

$$\text{Max } Z_1 = \sqrt[n-1]{\prod_{j=2}^{n}\left(\mathrm{SKM}_{(j)u} - \mathrm{SKM}_{(j-1)u} + \Delta\right)}, \tag{10}$$

where $\mathrm{SKM}_{(j)u}$ is the slack of the job that ranks the jth according to 2f-TNFSMCT. On the other hand, in 2f-TNFSVCT, the following objective function is to be optimized:

$$\text{Max } Z_2 = \sqrt[n-1]{\prod_{j=2}^{n}\left(\mathrm{SKV}_{(j)u} - \mathrm{SKV}_{(j-1)u} + \Delta\right)}, \tag{11}$$

where $\mathrm{SKV}_{(j)u}$ is the slack of the job that ranks the jth according to 2f-TNFSVCT. To fuse the two rules into a single one, that is, 2f-biFS, a natural way is to maximize the weighted geometric mean of the two objective functions, subject to several linear and nonlinear constraints:

$$\text{Max } Z_3 = Z_1^{\omega} Z_2^{1-\omega} \tag{12}$$

subject to

$$Z_1 = \sqrt[n-1]{\prod_{j=2}^{n}\left(\mathrm{SKM}_{(j)u} - \mathrm{SKM}_{(j-1)u} + \Delta\right)}$$

$$Z_2 = \sqrt[n-1]{\prod_{j=2}^{n}\left(\mathrm{SKV}_{(j)u} - \mathrm{SKV}_{(j-1)u} + \Delta\right)}$$

$$\mathrm{SKM}_{ju} = \left(\frac{\lambda\beta}{(n-1)\left(\mathrm{RCTE}_{ju} - \min\left(\mathrm{RCTE}_{ju}\right)\right)} \right)^{\xi} \cdot \left(\frac{j}{\lambda} - \mathrm{RCTE}_{ju} + \zeta\left(\mathrm{RCTE}_{ju} - \frac{1}{\lambda}\right)\right)$$

$$\mathrm{SKV}_{ju} = \left(\frac{\beta}{\alpha\left(\mathrm{RCTE}_{ju} - \min\left(\mathrm{RCTE}_{ju}\right)\right)} \right)^{\xi} \cdot \left(R_j - \mathrm{RCTE}_{ju} + \zeta\left(\mathrm{RCTE}_{ju} - \min\left(R_j\right)\right)\right)$$

$$\xi = \zeta, \tag{13}$$

$$\xi = \zeta^k, \quad k \geq 0, \tag{14}$$

or

$$\xi = \frac{\ln(1 + \zeta)}{\ln 2}, \quad 0 \leq \xi, \ \zeta \leq 1. \tag{15}$$

This model is obviously a NLP problem. Since ξ and ζ are limited to be within [0 1], the following exhaustive search algorithm can be used to reach the optimal solution of the NLP problem in an effective way.

Step 1. Choose a model from (6)~(8).

Step 2. Determine the weight ω.

Step 3. Let $\xi = 0$. $Z_{\max} = 0$

Step 4. Calculate ζ.

Step 5. Calculate the slack of each job according to 2f-TNFSMCT.

Step 6. Calculate Z_1.

Step 7. Calculate the slack of each job according to 2f-TNFSVCT.

Step 8. Calculate Z_2.

Step 9. Calculate Z_3. If Z_3 is greater than $Z_{\max}$, then $Z_{\max} = Z_3$.

Step 10. $\xi = \xi + 0.01$. If ξ is greater than 1, go to Step 11; otherwise, return to Step 4.

Step 11. Search the neighborhood of the current optimal solution to fine-tune the valueof ξ.

An example is provided in Table 5 to illustrate the procedure mentioned above, in which $\alpha = 734$, $\beta = 1836$, and $\lambda = 1.18$. If the nonlinear model in (7) is referred to and $k = 2$; $\omega = 0.6$, then the optimal objective function value is $Z_3^* = 47.1$ when $(\xi^*, \zeta^*) = (0.000016, 0.004)$.

Table 5: A demonstrative example.

#	R_j (hrs)	j	RCTE_{ju} (hrs)
1	102	159	1399
2	756	37	1127
3	826	37	1223
4	652	86	1822
5	208	55	530
6	783	84	2040
7	800	96	2366
8	478	52	942
9	469	65	1116
10	699	32	995
11	836	85	2151
12	497	45	883
13	596	101	2047
14	798	34	1146
15	197	79	743
16	804	85	2092
17	163	78	647
18	457	44	810
19	523	100	1851

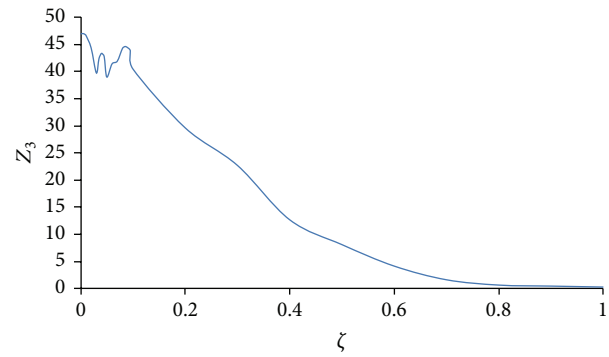


Figure 4: The objective function values associated with various values of ζ.

A parametric analysis was also performed. The objective function values associated with various values of ζ ($\xi = \zeta^2$) are shown in Figure 4. In theory, Z_3 converges to a small positive value (Δ).

The remaining cycle time of a job is an important input to the 2f-biFS rule. To estimate it, the FCM-BPN ensemble approach [30] was modified, and a new aggregation mechanism was proposed, as described in the next section.

2.3. The FCM Approach. FCM has been applied to a variety of purposes in a wafer fabrication factory. For instance, Liu and Chen [38] proposed a modified FCM algorithm along with a quality index (ψ) to cluster the characteristic values of low-yield wafers. Wang [39] proposed a hybrid scheme combining the entropy FCM (EFCM) with spectral clustering to denoise the noisy wafer map and to extract meaningful defect clusters. In the proposed methodology, the remaining cycle

times of all jobs in front of a machine must be estimated before determining the order of these jobs. In addition, job classification is considered to be beneficial to the accuracy of the remaining cycle time estimation [30, 36–39]. For these reasons, in the FCM-BPN ensemble approach, jobs are classified into K clusters using FCM. The use of FCM has the following advantages.

(1) Classification is a subjective concept. Therefore, the absolute job classification may not be correct. In a fuzzy classification method like FCM, a job belongs to more than one cluster with different degrees, which provides a solution to this problem. The FCM-BPN ensemble approach considers the estimates from all of the clusters and may be more robust than the FCM-BPN without ensemble.

(2) In a crisp clustering method, it is possible that some clusters have a very few examples. In contrast, in FCM all examples belong to a cluster with different degrees, which provides a solution to this problem.

The objective function of FCM is to minimize the weighted sum of squared error:

$$\text{Min} \sum_{k=1}^{K} \sum_{j=1}^{n} \mu_{j(k)}^{m} e_{j(k)}^{2}, \tag{16}$$

where K is the required number of clusters; n is the number of jobs; $\mu_{j(k)}$ indicates the membership that job j belongs to cluster k; $e_{j(k)}$ measures the distance from job j to the centroid of cluster k; $m \in [1, \infty)$ is a parameter to adjust the fuzziness and is usually set to 2. The procedure of FCM starts from the normalization of data. The (normalized) attributes of job j are placed in vector $\mathbf{x_j} = [x_{jp}]$. Subsequently, an initial guess of the clustering results is generated. The performance of FCM is highly sensitive to the initial guess. After each round of clustering, the centroid of each cluster is updated as follows:

$$\begin{aligned} \overline{x}_{(k)} &= \left\{\overline{x}_{(k)p}\right\}; \quad p = 1 \sim P, \\ \overline{x}_{(k)p} &= \frac{\sum_{j=1}^{n} \mu_{j(k)}^{m} x_{jp}}{\sum_{j=1}^{n} \mu_{j(k)}^{m}} \end{aligned} \tag{17}$$

in which

$$\mu_{j(k)} = \frac{1}{\sum_{q=1}^{K} \left(e_{j(k)}/e_{j(q)}\right)^{2/(m-1)}}, \tag{18}$$

$$e_{j(k)} = \sqrt{\sum_{\text{all } p} \left(x_{jp} - \overline{x}_{(k)p}\right)^{2}}, \tag{19}$$

where $\overline{x}_{(k)}$ is the centroid of cluster k. $\mu_{j(k)}$ and $e_{j(k)}$ are also updated at the same time. The iterative process of job clustering continues until the clustering results converge:

$$\max_{k} \max_{j} \left|\mu_{j(k)}^{(t)} - \mu_{j(k)}^{(t-1)}\right| < d, \tag{20}$$

where d is a real number representing the threshold for the convergence of membership. A problem of FCM is that the number of clusters must be decided in advance. To this end, the separate distance test (S test) proposed by Xie and Beni [40] is applicable:

$$S = \frac{J_m}{n \times e_{\min}^{2}}, \tag{21}$$

where

$$J_m = \sum_{k=1}^{K} \sum_{j=1}^{n} \mu_{j(k)}^{m} e_{j(k)}^{2}, \tag{22}$$

$$e_{\min}^{2} = \min_{k1 \neq k2} \left(\sum_{\text{all } p} \left(\overline{x}_{(k1)p} - \overline{x}_{(k2)p}\right)^{2} \right), \tag{23}$$

$$K \in \mathrm{Z}^{+}. \tag{24}$$

K is the number of categories. According to (22) and (21), J_m is a function of K, and S is a function of J_m. So, we can try several values of K to minimize S. In fact, the K value minimizing S determines the suitable number of clusters. After FCM, the original digital manufacturing model has been split into several smaller ones that are easier to deal with.

An example is provided in Table 6 to illustrate the application of FCM. All decision and response variables have been normalized into [0.1 0.9] to consider the situation that the future value may be greater/smaller than all the historical values. The Fuzzy Logic Toolbox of MATLAB is used to implement the FCM approach.

The results of the S test were summarized in Table 7. In this case, the optimal number of clusters was 3. The clustering results when $K = 3$ are shown in Table 8. If each job is classified into the cluster to which the membership is the highest, then the classification results are as shown in Table 9.

2.4. The BPN Approach. After clustering, a BPN is constructed for each cluster. A portion of the jobs in each cluster is input as the "training examples" to the BPN to determine the parameter values. The configuration of the BPN is established as follows. The BPN is a three-layer multiple-input single-output (MISO) system. Inputs are the normalized values of the P parameters associated with a job. There is only one hidden layer with $2P$ neurons. The output from the BPN is the normalized value of the remaining cycle time estimate. The activation function used in each layer is Log Sigmoid function (see Figure 5). The Levenberg-Marquardt algorithm is applied to train the BPN because of its efficiency. BPN training using the Levenberg-Marquardt algorithm has been stated in many past studies in this field [29] and therefore will not be repeated here.

In the previous example, take cluster 1 as an example. There are 16 jobs in this cluster. Split them into two parts: the training data (the first 12 jobs) and the testing data (the remaining jobs) and then construct a BPN to estimate the cycle time of jobs in this cluster. The Neural Network Toolbox of MATLAB is used to implement the BPN approach. The estimation results were summarized in Table 10. The estimation performance was basically very good. For the testing data, however, the estimation error was relatively large.

Table 6: An illustrative example.

j	u	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	RCTE$_{ju}$
1	197	0.300	0.485	0.900	0.367	0.367	0.162	0.667
2	192	0.500	0.604	0.900	0.367	0.367	0.362	0.698
3	188	0.300	0.248	0.100	0.456	0.500	0.163	0.758
4	202	0.500	0.159	0.900	0.100	0.233	0.203	0.738
5	187	0.500	0.604	0.100	0.544	0.567	0.349	0.675
6	184	0.500	0.396	0.900	0.722	0.700	0.279	0.703
7	197	0.300	0.426	0.900	0.633	0.367	0.314	0.719
8	184	0.300	0.574	0.900	0.811	0.567	0.314	0.727
9	178	0.500	0.811	0.900	0.900	0.633	0.314	0.670
10	212	0.300	0.752	0.900	0.633	0.567	0.106	0.783
11	177	0.300	0.900	0.900	0.722	0.633	0.106	0.697
12	186	0.300	0.426	0.900	0.544	0.500	0.106	0.736
13	199	0.300	0.396	0.900	0.456	0.567	0.106	0.713
14	195	0.300	0.159	0.900	0.544	0.567	0.100	0.806
15	223	0.500	0.189	0.100	0.278	0.367	0.291	0.858
16	223	0.300	0.781	0.100	0.456	0.300	0.168	0.900
17	206	0.100	0.722	0.900	0.189	0.100	0.105	0.722
18	212	0.100	0.367	0.900	0.367	0.433	0.105	0.833
19	168	0.500	0.544	0.900	0.278	0.167	0.351	0.614
20	200	0.500	0.426	0.900	0.278	0.167	0.351	0.678
21	140	0.100	0.841	0.900	0.367	0.167	0.142	0.445
22	156	0.300	0.130	0.100	0.456	0.167	0.162	0.548
23	206	0.300	0.396	0.900	0.633	0.633	0.123	0.820
24	152	0.300	0.515	0.900	0.811	0.767	0.123	0.506
25	187	0.100	0.544	0.100	0.722	0.700	0.123	0.633
26	185	0.500	0.781	0.900	0.722	0.433	0.319	0.634
27	137	0.500	0.663	0.900	0.722	0.567	0.260	0.423
28	151	0.700	0.396	0.900	0.811	0.633	0.620	0.486
29	148	0.500	0.100	0.900	0.722	0.567	0.250	0.402
30	173	0.300	0.426	0.900	0.722	0.700	0.216	0.667
31	161	0.300	0.307	0.900	0.722	0.767	0.158	0.486
32	108	0.500	0.396	0.900	0.811	0.700	0.264	0.155
33	115	0.300	0.870	0.900	0.900	0.900	0.302	0.164
34	100	0.300	0.485	0.900	0.633	0.633	0.337	0.150
35	97	0.900	0.396	0.900	0.900	0.633	0.900	0.100

The estimation accuracy can be evaluated with the following indexes:

$$\text{MAE} = \frac{\sum_{j=1}^{n} \left|\text{CT}_j - \text{CTE}_j\right|}{n},$$

$$\text{MAPE} = \frac{\sum_{j=1}^{n} \left|\text{CT}_j - \text{CTE}_j\right| / \text{CT}_j}{n} \cdot 100\%, \quad (25)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{n} \left(\text{CT}_j - \text{CTE}_j\right)^2}{n}}.$$

a_j and f_j denote the actual value and forecast of job j, respectively; n is the total number of data. In this example,

MAE = 43 (hrs),

MAPE = 3.7%,

RMSE = 62 (hrs).

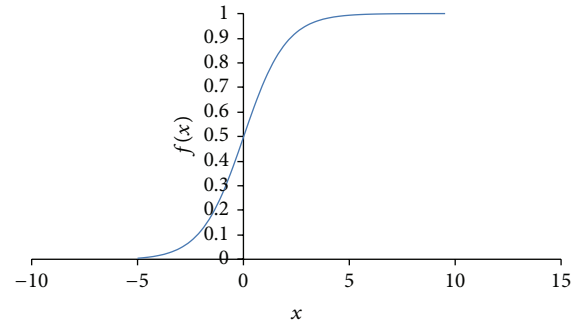


Figure 5: The Log Sigmoid function.

The performance of the FCM-BPN approach was also compared with those of BPN trained using the Levenberg-Marquardt algorithm and FCM-BPN trained using the gradient descent algorithm. The results are shown in Table 11.

Table 7: The results of the S test.

Number of clusters (K)	J_m	$e^2_{\min}$	S
2	0.0739	0.0027	0.7946
3	0.0444	0.0028	0.4586
4	0.0316	0.0014	0.6298
5	0.0247	0.0014	0.4882
6	0.0205	0.0009	0.6336
7	0.0175	0.0005	1.1102
8	0.0154	0.0003	1.2577

Table 8: The clustering results by FCM.

j	$\mu_{j(1)}$	$\mu_{j(2)}$	$\mu_{j(3)}$
1	0.91	0.061	0.03
2	0.698	0.21	0.092
3	0.06	0.046	0.894
4	0.559	0.22	0.221
5	0.118	0.118	0.764
6	0.262	0.664	0.073
7	0.721	0.218	0.062
8	0.298	0.629	0.073
9	0.244	0.654	0.101
10	0.497	0.396	0.107
11	0.365	0.508	0.127
12	0.802	0.152	0.047
13	0.766	0.173	0.061
14	0.567	0.284	0.149
15	0.139	0.097	0.764
16	0.186	0.137	0.678
17	0.574	0.224	0.202
18	0.691	0.179	0.13
19	0.635	0.22	0.145
20	0.649	0.202	0.149
21	0.521	0.296	0.182
22	0.164	0.122	0.714
23	0.546	0.353	0.101
24	0.173	0.766	0.061
25	0.18	0.195	0.624
26	0.355	0.55	0.095
27	0.134	0.828	0.038
28	0.264	0.6	0.137
29	0.332	0.527	0.141
30	0.275	0.661	0.064
31	0.253	0.659	0.088
32	0.212	0.682	0.106
33	0.246	0.608	0.146
34	0.26	0.632	0.107
35	0.295	0.487	0.218

2.5. Aggregating the Estimates from BPNs. In past studies, a job is usually classified into the cluster with the highest membership. However, that makes FCM not different from crisp classifiers. To tackle this problem, Chen [30] applied the BPNs of all clusters to estimate the cycle time of a job and used a BPN to aggregate these estimates. However, the BPN aggregator is very sensitive and may lead to unexpected results for untrained data. In addition, Chen [30] also showed that the aggregation performance of simple linear combination was even worse. For these reasons, a new aggregation mechanism is proposed as follows.

Table 9: The classification results (the highest membership principle).

Cluster	Jobs
1	6, 8, 9, 10, 11, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34
2	3, 5, 15, 16, 22, 25
3	1, 2, 4, 7, 12, 13, 17, 18, 19, 20, 21

Table 10: The estimation results (cluster 1).

j	CT_j	CTE_j	$\lvert\mathrm{CT}_j - \mathrm{CTE}_j\rvert$
6	1227	1226	1.6
8	1215	1230	15.3
9	1228	1231	2.5
10	1266	1260	5.4
11	1285	1268	17.0
23	1280	1238	42.1
24	1286	1258	28.0
26	1214	1224	9.6
27	1251	1227	23.9
28	1222	1193	29.4
29	1187	1204	17.5
30	1205	1235	30.4
31	1120	1240	120.1
32	1133	1220	88.0
33	1130	1263	133.2
34	1113	1238	124.5

Table 11: Comparison of the estimation performances of various approaches.

Performance measure	BPN (the Levenberg-Marquardt algorithm)	FCM-BPN (the gradient descent algorithm)	FCM-BPN (the Levenberg-Marquardt algorithm)
MAE	47 (hrs)	79	43
MAPE	4.1%	6.7%	3.7%
RMSE	70 (hrs)	98	62

According to (18), μ_{jk} is inversely proportional to $e^{2/(m-1)}_{j(k)}$:

$$\mu_{j(k)} \propto \frac{1}{e^{2/(m-1)}_{j(k)}}. \tag{26}$$

Therefore,

$$e_{j(k)} \propto \sqrt[2/(m-1)]{\frac{1}{\mu_{j(k)}}}. \tag{27}$$

Further, according to (19), the error is proportional to the distance to the centroid. For this reason, a natural way to aggregate the estimates from the BPNs is

$$\mathrm{RCTE}_{ju} = \frac{\sum_{k=1}^{K} \sqrt[2/(m-1)]{1/\mu_j(k)} \cdot \mathrm{RCTE}_{ju}(k)}{\sum_{k=1}^{K} \sqrt[2/(m-1)]{1/\mu_j(k)}}, \tag{28}$$

where $\mathrm{RCTE}_{ju}(k)$ is the remaining cycle time of job j estimated by the BPN of cluster k.

Table 12: The estimation results after aggregation (cluster 1).

j	CT_j	CTE_j	$\lvert CT_j - CTE_j \rvert$
6	1227	1201	26.3
8	1215	1199	16.4
9	1228	1170	58.2
10	1266	1228	38.0
11	1285	1222	62.8
23	1280	1203	77.1
24	1286	1240	45.7
26	1214	1187	26.7
27	1251	1195	56.4
28	1222	1158	64.0
29	1187	1191	3.6
30	1205	1176	28.8
31	1120	1147	27.0
32	1133	1103	30.0
33	1130	1172	42.0
34	1113	1158	45.0

In the previous example, after aggregation, the estimation results are shown in Table 12. The estimation accuracy, measured in terms of MAE, for example, was slightly improved from 43 hours to 40.5 hours. It is worth noting that the estimation error was significantly reduced for the untrained (testing) data, which supported the robustness of the aggregation mechanism. In Table 13, the performances of some aggregation mechanisms were also compared. In linear combination, the weighted sum of the estimates from all of the BPNs was obtained, in which the weight was equal to the membership of belonging to a cluster. According to the experimental results, although the BPN aggregation mechanism achieved the best estimation accuracy, it may generate a considerable deviation for untrained (testing) data. In contrast, the proposed aggregation mechanism did not have this problem, and the estimation accuracy was also quite good.

3. Experiments

Twelve scheduling policies, including FIFO, EDD, CR, SRPT, FSVCT, FSMCT, DBD [11], NFS [23], 2f-TNFSMCT, 2f-TNFSVCT, the slack-diversifying rule [29], and the proposed FCM-BPN-ensemble-2f-biFS, were applied to schedule the target wafer fabrication factory. In total, the data of 1000 jobs of five major cases have been collected and were separated by their product types and priorities.

In FIFO, jobs were sequenced on each machine first by their priorities, then by their arrival times at the machine. In EDD, jobs were sequenced first by their priorities, then by their due dates that were equal to the multiple of the total processing times plus the release time:

$$DD_j = R_j + c \cdot TPT_j, \tag{29}$$

where c is called the cycle time multiplier and is determined based on the cycle time statistics. In CR, jobs were sequenced first by their priorities, then by their critical ratios that are defined as

$$CR_j = \frac{DD_j - t}{RPT_{ju}}. \tag{30}$$

In SRPT, jobs were sequenced first by their priorities, then by their remaining processing times. In FSVCT and FSMCT, jobs were sequenced on each machine first by their priorities and then by their slack values, which were determined by (1) and (2). To assist this, the remaining cycle time statistics have been collected from the historical data. The operation of NFS is similar, in which the slack of a job is calculated in the following way:

$$SK_{ju} = \left(\frac{R_j - \min R_j}{\max R_j - \min R_j}\right)^{\xi} \cdot \left(\frac{j-1}{n-1}\right)^{1-\xi} \times \left(\frac{RCTE_{ju} - \min RCTE_{ju}}{\max RCTE_{ju} - \min RCTE_{ju}}\right)^{-1}, \tag{31}$$

where ξ ranges between 0 and 1 and will be optimized by trying various values in the experiment. In DBD, workstations are first classified into four categories, and then CR + FIFO, the shortest processing time until the next bottleneck (SPNB) + CR + FIFO, shortest processing time (SPT) + CR + FIFO, and CR + FIFO are applied to sequence jobs in these categories, respectively. In the slack-diversifying rule [29], the weighted sum of the standard deviations of slacks calculated by 2f-TNFSMCT and 2f-TNFSVCT was obtained, and then jobs were sequenced according to this value.

Two performance measures, the average cycle time and cycle time standard deviation of each case, achieved by the twelve scheduling policies were observed and summarized in Tables 14 and 15.

According to the experimental results, the following points can be made.

(1) For the average cycle time, the proposed FCM-BPN-ensemble-2f-biFS outperformed the eleven existing policies. The most significant advantage was over the 2f-TNFSVCT rule, which is about 28% on average.

(2) On the other hand, the proposed FCM-BPN-ensemble-2f-biFS also surpassed the existing rules, especially FSMCT, in reducing cycle time standard deviation. The average advantage was up to 61%.

(3) Both the slack-diversifying rule [29] and the proposed FCM-BPN-ensemble-2f-biFS achieved very good

Table 13: The performances of various aggregation mechanisms (cluster 1).

MAE (hrs)	Without aggregation	Linear combination	BPN	The proposed mechanism
Training data	18.6	60.0	12.7	42.0
Testing data	116.5	147.8	90.9	36.0
Both	43.0	81.9	32.3	40.5

Table 14: The performances of various approaches in the average cycle time.

Avg. cycle time (hrs)	Case I	Case II	Case III	Case IV	Case V
FIFO	1254	400	317	1278	426
EDD	1098	347	302	1431	436
CR	1145	354	299	1501	443
SRPT	948	350	308	1737	457
FSMCT	1313	347	293	1851	470
FSVCT	1014	382	315	1672	475
NFS	1456	407	321	1452	421
DBD	1031	349	297	1501	463
2f-TNFSMCT	1369	379	306	1361	399
2f-TNFSVCT	1465	416	318	1551	500
The slack-diversifying rule	1076	289	269	1132	388
FCM-BPN-ensemble-2f-biFS	987	276	254	1121	376

Table 15: The performances of various approaches in cycle time standard deviation.

Cycle time standard deviation (hrs)	Case I	Case II	Case III	Case IV	Case V
FIFO	55	24	25	87	51
EDD	128	25	23	50	61
CR	70	30	18	57	52
SRPT	248	31	22	106	53
FSMCT	419	33	16	129	104
FSVCT	280	37	27	201	77
NFS	87	49	19	44	47
DBD	136	25	19	77	29
2f-TNFSMCT	75	37	17	47	19
2f-TNFSVCT	38	38	29	33	24
The slack-diversifying rule	86	26	15	54	21
FCM-BPN-ensemble-2f-biFS	83	21	16	55	19

scheduling performances, which supported slack diversifying to be a viable strategy for improving the performances of similar rules.

(4) In addition, the performance of the proposed FCM-BPN-ensemble-2f-biFS was also better than the slack-diversifying rule [29]. This showed that the treatments taken in this study, including the new way of slack diversification and forming the BPN ensemble, were indeed effective. To differentiate their effects, additional experiments have been conducted, and the results were shown in Figure 6. Obviously, forming the BPN ensemble was more effective than the new way of slack diversification. That was reasonable because in theory slack diversification has its limits.

(5) If the cycle time is long, the remaining cycle time will be much longer than the remaining processing time, which leads to the ineffectiveness of SRPT. As a result, SRPT performed poor in reducing the average cycle times of such cases, such as case IV. FSMCT has similar problems.

(6) In contrast, the performances of EDD and CR were satisfactory for cases with short cycle times because it is less likely that the cycle time (multiplier) will deviate from the estimated value.

(7) We also compared the performances of different fusion mechanisms. To this end, the linear fusion in which the slack values derived by the two rules were simply added was also tested. The results are shown in Figure 7. Obviously, the effects of the linear fusion mechanism on the average cycle time were poor, and certainly the proposed geometric fusion mechanism could be used as an alternative.

4. Conclusions and Directions for Future Research

To further improve the performance of job scheduling in a wafer fabrication factory, the slack-diversifying rule used by Chen [29] is modified, and the FCM-BPN-ensemble-2f-biFS was proposed in this study. In the proposed methodology, two existing rules, 2f-TNFSMCT and 2f-TNFSVCT, are fused by maximizing the geometric mean of the neighboring distances of slacks. In addition, to enhance the accuracy of estimating the remaining cycle time, the FCM-BPN ensemble approach is applied with a novel aggregation mechanism that considers the operations of FCM and will not overreact for untrained data.

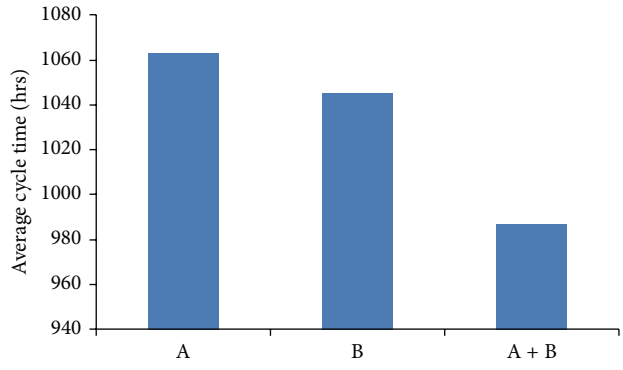


Figure 6: The effects of different treatments (A: the new way of slack diversification; B: forming the BPN ensemble).

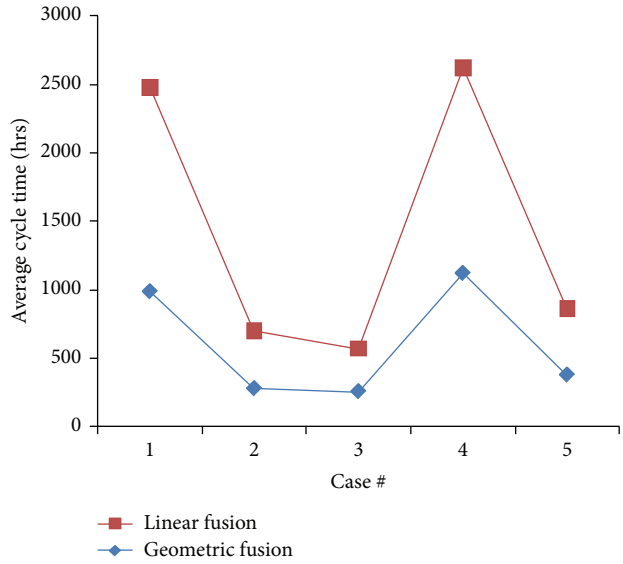


Figure 7: The performance of different fusion mechanisms.

The effectiveness of the proposed FCM-BPN-ensemble-2f-biFS was evaluated with some experiments. According to the experimental results, the following happened.

(1) The proposed FCM-BPN-ensemble-2f-biFS outperformed the eleven existing methods/rules in reducing the average cycle time and cycle time standard deviation simultaneously.

(2) To diversify the slacks of jobs, maximizing the geometric mean of the neighboring distances of slacks is a better choice than maximizing the standard deviation of slacks.

(3) Improving the accuracy of the remaining cycle time estimation was shown to be a more effective way to the improvement of the scheduling performance, while slack diversification has its limits.

However, this study has its limits that can only be solved by applying it to an actual wafer fabrication factory. In addition, developing different versions of the proposed rule for bottleneck and nonbottleneck machines can be considered in future studies. Further, to optimize the values of the parameters, other search algorithms that reach near-optimal solutions in less time are to be developed.

Abbreviations

j: Job index; $j = 1 \sim n$
u: Step number
CR_{ju}: The critical ratio of job j at step u; $j = 1 \sim n$
CT_j: The cycle time of job j
CTE_j: The estimated cycle time of job j
DD_j: The due date of job j
R_j: The release time of job j
$RCTE_{ju}$: The estimated remaining cycle time of job j from step u
RPT_{ju}: The remaining processing time of job j from step u
SCT_{ju}: The step cycle time of job j until step u
SK_{ju}: The slack of job j at step u
SKM_{ju}: The slack of job j at step u in FSMCT and its variants
SKV_{ju}: The slack of job j at step u in FSVCT and its variants
t: The current time
TPT_j: The total processing time of job j
λ: Mean release rate
h_l: The output from hidden-layer node l, $l = 1 \sim L$
w_{pl}^h: The connection weight between input node p and hidden-layer node l, $p = 1 \sim P$; $l = 1 \sim L$
w_l^o: The connection weight between hidden-layer node l and the output node
x_p: Inputs to the BPN, $p = 1 \sim P$
θ_l^h: The threshold on hidden-layer node l
θ^o: The threshold on the output node.

Acknowledgment

This work was supported by the National Science Council of Taiwan.

References

[1] C. N. Wang and C. H. Wang, "A simulated model for cycle time reduction by acquiring optimal lot size in semiconductor manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 34, no. 9-10, pp. 1008–1015, 2007.

[2] T. Chen and Y. C. Lin, "A fuzzy-neural fluctuation smoothing rule for scheduling jobs with various priorities in a miconductor manufacturing factory," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 17, no. 3, pp. 397–417, 2009.

[3] T. Chen and Y. C. Wang, "A nonlinear scheduling rule incorporating fuzzy-neural remaining cycle time estimator for scheduling a semiconductor manufacturing factory—a simulation study," *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 110–121, 2009.

[4] D. A. Koonce and S. C. Tsai, "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule," *Computers and Industrial Engineering*, vol. 38, no. 3, pp. 361–374, 2000.

[5] B. W. Hsieh, C. H. Chen, and S. C. Chang, "Scheduling semiconductor wafer fabrication by using ordinal optimization-based simulation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 599–608, 2001.

[6] H. J. Yoon and W. Shen, "A multiagent-based decision-making system for semiconductor wafer fabrication with hard temporal constraints," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 83–91, 2008.

[7] Y. Harrath, B. Chebel-Morello, and N. Zerhouni, "A genetic algorithm and data mining based meta-heuristic for job shop scheduling problem," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 7, pp. 280–285, October 2002.

[8] K. Sourirajan and R. Uzsoy, "Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication," *Journal of Scheduling*, vol. 10, no. 1, pp. 41–65, 2007.

[9] A. K. Gupta and A. I. Sivakumar, "Job shop scheduling techniques in semiconductor manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 27, no. 11-12, pp. 1163–1169, 2006.

[10] K. Altendorfer, B. Kabelka, and W. Stöcher, "A new dispatching rule for optimizing machine utilization at a semiconductor test field," in *Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC '07)*, pp. 188–193, Stresa, Italy, June 2007.

[11] H. Zhang, Z. Jiang, and C. Guo, "Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology," *International Journal of Advanced Manufacturing Technology*, vol. 41, no. 1-2, pp. 110–121, 2009.

[12] Z. Cao, Y. Peng, and Y. Wang, "A drum-buffer-rope based scheduling method for semiconductor manufacturing system," in *Proceedings of the 7th IEEE International Conference on Automation Science and Engineering (CASE '11)*, pp. 120–125, Trieste, Italy, August 2011.

[13] Y. H. Lee, C. T. Chang, D. S. H. Wong, and S. S. Jang, "Petri-net based scheduling strategy for semiconductor manufacturing processes," *Chemical Engineering Research and Design*, vol. 89, no. 3, pp. 291–300, 2011.

[14] I. L. Wang, Y. C. Wang, and C. W. Chen, "Scheduling unrelated parallel machines in semiconductor manufacturing by problem reduction and local search heuristics," *Flexible Services and Manufacturing Journal*, vol. 25, no. 3, pp. 343–366, 2013.

[15] T. Chen, "Fuzzy-neural-network-based fluctuation smoothing rule for reducing the cycle times of jobs with various priorities in a wafer fabrication plant: a simulation study," *Proceedings of the Institution of Mechanical Engineers B*, vol. 223, no. 8, pp. 1033–1043, 2009.

[16] H. Hu, L. Zhen, Z. Sun, and H. Zhang, "A multi-stage fluctuation smoothing method for multiple bottlenecks in wafer fabrication," *International Journal of Advanced Manufacturing Technology*, 2013.

[17] C. C. Chern and Y. L. Liu, "Family-based scheduling rules of a sequence-dependent wafer fabrication system," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 1, pp. 15–24, 2003.

[18] T. Chen, "A tailored non-linear fluctuation smoothing rule for semiconductor manufacturing factory scheduling," *Proceedings of the Institution of Mechanical Engineers I*, vol. 223, no. 2, pp. 149–160, 2009.

[19] T. Chen, "Intelligent scheduling approaches for a wafer fabrication factory," *Journal of Intelligent Manufacturing*, vol. 23, no. 3, pp. 897–911, 2012.

[20] R. M. Dabbas and J. W. Fowler, "A new scheduling approach using combined dispatching criteria in wafer fabs," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, pp. 501–510, 2003.

[21] R. M. Dabbas, H. N. Chen, J. W. Fowler, and D. Shunk, "A combined dispatching criteria approach to scheduling semiconductor manufacturing systems," *Computers and Industrial Engineering*, vol. 39, no. 3-4, pp. 307–324, 2001.

[22] T. C. Chiang, Y. S. Shen, and L. C. Fu, "A new paradigm for rule-based scheduling in the wafer probe centre," *International Journal of Production Research*, vol. 46, no. 15, pp. 4111–4133, 2008.

[23] T. Chen and Y. C. Wang, "A bi-criteria nonlinear fluctuation smoothing rule incorporating the SOM-FBPN remaining cycle time estimator for scheduling a wafer fab-a simulation study," *International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5-8, pp. 709–721, 2010.

[24] T. Chen, Y. C. Wang, and Y. C. Lin, "A bi-criteria four-factor fluctuation smoothing rule for scheduling jobs in a wafer fabrication factory," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 10, pp. 4289–4303, 2010.

[25] Y. D. Kim, D. H. Lee, J. U. Kim, and H. K. Roh, "A simulation study on lot release control, mask scheduling, and batch scheduling in semiconductor wafer fabrication facilities," *Journal of Manufacturing Systems*, vol. 17, no. 2, pp. 107–117, 1998.

[26] R. M. Dabbas, J. W. Fowler, D. A. Rollier, and D. McCarville, "Multiple response optimization using mixture-designed experiments and desirability functions in semiconductor scheduling," *International Journal of Production Research*, vol. 41, no. 5, pp. 939–961, 2003.

[27] X. Li and O. Sigurdur, "Data mining for best practices in scheduling data," in *Proceedings of the IIE Annual Conference and Exhibition*, p. 853, May 2004.

[28] T. Chen, "An optimized tailored nonlinear fluctuation smoothing rule for scheduling a semiconductor manufacturing factory," *Computers and Industrial Engineering*, vol. 58, no. 2, pp. 317–325, 2010.

[29] T. Chen, "A nonlinear programming and artificial neural network approach for optimizing the performance of a job dispatching rule in a wafer fabrication factory," *Applied Computational Intelligence and Soft Computing*, vol. 2012, Article ID 471973, 9 pages, 2012.

[30] T. Chen, "Incorporating fuzzy c-means and a back-propagation network ensemble to job completion time prediction in a semiconductor fabrication factory," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2153–2168, 2007.

[31] S. Z. Zhao, Y. Cao, and Q. F. Tian, "Urban rail transit scheduling decision based on information fusion," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 41, supplement 1, pp. 85–88, 2011.

[32] G. Lebret, G. Yao, M. Ait-Ahmed, and T. Tang, "A gain-scheduling and intelligence fusion method for fault-tolerant control," *IFAC Proceedings*, vol. 6, no. 1, pp. 1258–1263, 2006.

[33] X. Li, H. Kang, and H. H. Chen, "Sensing workload scheduling in hierarchical sensor networks for data fusion applications," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '07)*, pp. 214–219, August 2007.

[34] S. C. H. Lu, D. Ramaswamy, and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pp. 374–388, 1994.

[35] S. D. Ashok and G. L. Samuel, "Least square curve fitting technique for processing time sampled high speed spindle data," *International Journal of Manufacturing Research*, vol. 6, no. 3, pp. 256–276, 2011.

[36] T. Chen, "Dynamic fuzzy-neural fluctuation smoothing rule for jobs scheduling in a wafer fabrication factory," *Proceedings of the Institution of Mechanical Engineers I*, vol. 223, no. 8, pp. 1081–1094, 2009.

[37] Y. C. Wang, T. Chen, and C. W. Lin, "A slack-diversifying nonlinear fluctuation smoothing rule for job dispatching in a wafer fabrication factory," *Robotics and Computer Integrated Manufacturing*, vol. 29, no. 3, pp. 41–47, 2013.

[38] S. F. Liu and F. L. Chen, "A data clustering model for wafer yield loss in semiconductor manufacturing," *Journal of the Chinese Institute of Industrial Engineers*, vol. 21, no. 4, pp. 328–338, 2004.

[39] C. H. Wang, "Recognition of semiconductor defect patterns using spatial filtering and spectral clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1914–1923, 2008.

[40] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 672404, 7 pages
http://dx.doi.org/10.1155/2013/672404

Research Article

Semiconductor Yield Forecasting Using Quadratic-Programming-Based Fuzzy Collaborative Intelligence Approach

Toly Chen and Yu-Cheng Wang

Department of Industrial Engineering and Systems Management, Feng Chia University, Taichung City 407, Taiwan

Correspondence should be addressed to Toly Chen; tcchen@fcu.edu.tw

Received 26 March 2013; Accepted 17 May 2013

Academic Editor: Yi-Chi Wang

Several recent studies have proposed fuzzy collaborative forecasting methods for semiconductor yield forecasting. These methods establish nonlinear programming (NLP) models to consider the opinions of experts and generate fuzzy yield forecasts. Such a practice cannot distinguish between the different expert opinions and can not easily find the global optimal solution. In order to solve some problems and to improve the performance of semiconductor yield forecasting, this study proposes a quadratic-programming- (QP-) based fuzzy collaborative intelligence approach.

1. Introduction

Yield is the proportion between goods and total products. Given the same inputs, the higher yield means more salable outputs that increase the revenues. Yield is a key performance measure in operations and has a dominant effect on manufacturing economics. Yield improvement is also a crucial work for a factory's competitiveness [1]. In addition, predicting the future yield to estimate the possible gains is also a basis for long-term production planning [2]. The ramping of a semiconductor manufacturing factory should occur quickly and then product yields should be maintained at a high level to maximize profits. This study is therefore committed to predicting the yield of a semiconductor product. Semiconductor products are focused on because of the extensive use in modern life, such as personal computer, laptop, cell phone, and others. According to Chen and Wang [3], methods for predicting the yield of a semiconductor product can be divided into two categories—macro yield modeling (MaYM) methods and micro yield modeling (MiYM) methods. This study belongs to the MaYM category, in which a learning model is used to predict the future yield of a semiconductor product. After conversion, some yield learning model can be solved using linear regression [3], which is convenient and explains why the related methods have been welcomed. However, there is much uncertainty in the yield learning process. To deal with this issue, several treatments have been taken in the literature (Figure 1).

(1) *Probabilistic/Stochastic Methods.* Methods of this sub-category assume that parameter distributions are known in advance to a certain degree, and these distributions can be modified in a Bayesian manner after actual values are observed. The works of Spence [4], Majd and Pindyck [5], Mazzola and McCardle [6], and Anderson [7] all belong to this sub-category.

(2) *Fuzzy Methods.* Methods of this sub-category fit the yield learning process with a possibility regression model, and yield forecasts are given in fuzzy values to consider the uncertainty [1, 3, 8–12]. In the work of Chen and Wang [3], the fuzzy yield learning curve was fitted by solving a linear programming (LP) problem. Wu et al. [12] established a fuzzy back propagation network (FBPN) to predict the yield of semiconductor. The inputs of this network include three categories: the physical parameters, electrical test parameters, and wafer defect parameters. Chiang and Hsieh [13] applied a similar fuzzy neural network

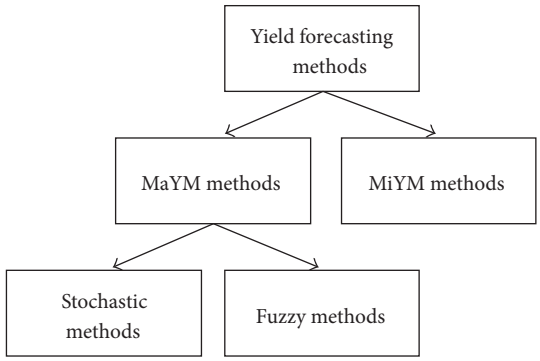


Figure 1: Classification of yield forecasting methods.

method, in which grey relational analysis was used to find the most influential parameters. Lin [11] considered the uncertainty of the defect clustering pattern and established a number of fuzzy rules to predict yield.

On the other hand, the concept of fuzzy collaborative intelligence was proposed by Pedrycz in 2003 [14] and has been applied to the forecasting of various properties, such as yield [8, 9], global CO_2 concentration [15], job cycle time [16], and others. Among them, yield forecasting received much attention and is the focus of this study. In [8], Chen and Lin proposed a fuzzy collaborative intelligence approach for forecasting the yield of a semiconductor product. In the fuzzy collaborative intelligence approach, a group of experts predict the future yield with a fuzzy number from different points of view. The forecasts by the experts are aggregated in such a way that both the precision and accuracy can be improved at the same time.

(1) Accuracy: the forecasted value should be as close as possible to the actual value.

(2) Precision: a narrow interval containing the actual value is established.

However, the fuzzy collaborative intelligence approach requires the solving of two nonlinear programming (NLP) problems, which is not easy. Furthermore, the existing optimization packages may not be able to guarantee the global optimality of the solutions. In addition, the views of the experts are not reflected properly on the objective function. In order to solve these two problems, the following treatments have been taken in this study.

(1) A new partitioning method has been proposed, which considers the effects on the objective function in partitioning the range of parameters.

(2) The nonlinear objective functions and constraints were converted into quadratic ones using several polynomial fitting techniques. Then, a variety of methods, such as the interior point method, the active set method, the augmented Lagrangian method, the conjugate gradient method, the gradient projection method, and the extension of the simplex algorithm, can be used to solve the quadratic programming (QP) problems.

The differences between the proposed methodology and the previous methods are summarized in Table 1.

The rest of this paper is organized as follows. Section 2 reviews the related literature. In Section 3, we proposed some countermeasures to solve the problems in the work of Chen and Lin [8], in order to propose a QP-based fuzzy collaborative forecasting method. Examples in the work of Chen and Lin [8] were used to assess the effectiveness of the proposed methodology, and to make a comparison with the NLP-based method. Finally, the conclusions of this study are made in Section 4.

2. Literature Review

Multiple analyses of a problem from diverse perspectives raise the chance that no relevant aspects of the problem will be ignored. In addition, as internet applications become widespread, dealing with disparate data sources is becoming more and more popular. Technical constraints, security issues, and privacy considerations often limit access to some sources. Therefore, the concepts of collaborative computing intelligence and collaborative fuzzy modeling have been proposed, and the certain so-called fuzzy collaborative systems are being established. In a fuzzy collaborative system, some experts, agents, or systems with various backgrounds are trying to achieve a common target. Since they have different knowledge and points of view, they may use various methods to model, identify, or control the common target. The key of such a system is that these experts, agents, or systems share and exchange their observations, settings, experiences, and knowledge with each other when achieving the common goal. This features the fuzzy collaborative system distinct from the ensemble of multiple fuzzy systems.

In the limited literature, fuzzy collaborative intelligence and systems have been successfully applied to collaborative clustering, group forecasting, agent negotiation, assessment and reputation, information filtering, robot control, intrusion detection, object tracking, and so forth. Applications of fuzzy collaborative intelligence in other fields remain to be investigated. Several studies have argued that for certain problems, a fuzzy collaborative intelligence approach is more precise, accurate, efficient, safe, and private than typical approaches.

Designing is a work in which the views may vary and therefore needs coordination or cooperation. Shai and Reich [17, 18] defined the concept of infused design as an approach for establishing effective collaboration between designers from different engineering fields. Ostrosi et al. [19] defined the concept of consensus as the overlapping of design clusters of different perspectives.

In demand forecasting, Lo et al. [20] believed that the demand plan has to be compromised and achieved through the collaborative efforts of all the demand network stakeholders. Büyüközkan and Vardaloğlu [21, 22] applied the fuzzy cognitive map method to the collaborative planning, forecasting, and replenishment of a supply chain. The initial

Table 1: The differences between the proposed methodology and the previous methods.

Method	Expert collaboration	Partitioning the ranges of parameters	Models to solve	Easiness to solve	Optimality of the solution
Chen and Wang [9]	No	No	LP	Very easy	Global optimal
Chen and Lin [8]	Yes	Uniform partitioning	NLP	Very difficult	Local optimal
Wu et al. [12]	No	No	FBPN	Very difficult	Local optimal
The proposed methodology	Yes	Considering the effects on the objective function	QP	Easy	Global optimal

values of the concepts and the connection weights of the fuzzy cognitive map are dependent on the subjective belief of the expert and can be modified after collaboration. Cheikhrouhou et al. [23] thought that collaboration is necessary because of the unexpected events that may occur in the future demand.

In the collaborative forecasting, according to Poler et al. [24], the comparison of collaboration methods and the proposing of software tools, especially regard forecasting methods for collaborative forecasting, are still lacking. Chen [16] put forward a fuzzy collaborative forecasting method to predict the job cycle time in a wafer fabrication factory, in which each expert used a fuzzy multiple linear regression equation to predict the cycle time of a job. The cycle time forecasts of different experts were aggregated using the hybrid fuzzy intersection (FI) and back propagation network (BPN) approach. A similar method has been used by Chen and Wang [15] to predict the global CO_2 concentration. The traditional method in this field is a simple time series analysis, but it is not accurate enough. Nonlinear methods, such as artificial neural networks [25, 26], may be more effective.

In all fuzzy collaborative intelligence methods, the consensus of results is being sought. Pedrycz and Rai [27] discussed the problem of collaborative data analysis by a group of agents having access to different parts of data and exchanging findings through their collaboration. A two-phase optimization procedure was established, so that the results of communication can be embedded into the local optimization results. Sometimes, however, the consensus may not exist. Therefore, Chen [28] defined the concept of partial consensus as the intersection of the views of some experts. In the work of Chen and Wang [9], an agent considered the settings of other agents, to optimize its own setting.

3. QP-Based Fuzzy Collaborative Intelligence Approach

According to Gruber [29], the general yield learning model of a semiconductor product is

$$Y_t = Y_0 e^{-b/t+r(t)}, \tag{1}$$

where b and Y_0 indicate the learning constant and the asymptotic yield, respectively. $b \geq 0$; $0 \leq Y_0, Y_t \leq 1$; t is time, $t = 1 \sim T$; $r(t)$ is a homoscedastic, serially noncorrelated error term. After converting to logarithms,

$$\ln Y_t = \ln Y_0 - \frac{b}{t} + r(t) = a - \frac{b}{t} - r(t), \tag{2}$$

where $a = \ln Y_0$. In Chen and Lin's fuzzy collaborative intelligence approach [8], to consider the uncertainty in yield learning, the parameters in (2) are given in triangular fuzzy numbers as follows:

$$\begin{aligned} \widetilde{Y}_t &= (Y_{t1}, Y_{t2}, Y_{t3}), \\ \widetilde{Y}_0 &= (Y_{01}, Y_{02}, Y_{03}), \\ \widetilde{b} &= (b_1, b_2, b_3), \\ \widetilde{a} &= (a_1, a_2, a_3), \end{aligned} \tag{3}$$

and the following two NLP problems are to be solved.

NLP problem I is

$$\text{Min } Z_1 = \sum_{t=1}^{T} (\ln Y_{t3} - \ln Y_{t1})^{o_k} \tag{4}$$

subject to

$$\begin{aligned} \ln y_t &\geq \ln Y_{t1} + s_k (\ln Y_{t2} - \ln Y_{t1}), \\ \ln y_t &\leq \ln Y_{t3} + s_k (\ln Y_{t2} - \ln Y_{t3}), \\ \ln Y_{t1} &= a_1 - \frac{b_3}{t}, \\ \ln Y_{t2} &= a_2 - \frac{b_2}{t}, \\ \ln Y_{t3} &= a_3 - \frac{b_1}{t}, \\ 0 &\leq a_1 \leq a_2 \leq a_3, \\ 0 &\leq b_1 \leq b_2 \leq b_3, \\ t &= 1 \sim T. \end{aligned} \tag{5}$$

NLP problem II is

$$\text{Max } Z_2 = \sqrt[m_k]{\frac{\sum_{t=1}^{T} s_t^{m_k}}{T}} \tag{6}$$

subject to

$$\sum_{t=1}^{T}\left(\ln Y_{t3}-\ln Y_{t1}\right)^{o_k}\le T\cdot {d_k}^{o_k},\tag{7}$$

$$\begin{aligned}
\ln y_t &\ge \ln Y_{t1}+s_t\left(\ln Y_{t2}-\ln Y_{t1}\right),\\
\ln y_t &\le \ln Y_{t3}+s_t\left(\ln Y_{t2}-\ln Y_{t3}\right),\\
\ln Y_{t1} &= a_1-\frac{b_3}{t},\\
\ln Y_{t2} &= a_2-\frac{b_2}{t},\\
\ln Y_{t3} &= a_3-\frac{b_1}{t},\\
0\le a_1 &\le a_2\le a_3,\\
0\le b_1 &\le b_2\le b_3,\\
t &= 1\sim T,
\end{aligned}\tag{8}$$

where $\widetilde{b}=(b_1,b_2,b_3)$ and $\widetilde{Y}_0=(Y_{01},Y_{02},Y_{03})$ indicate the learning constant and the asymptotic yield. t is time, $t=1\sim T$; o_k indicates the sensitivity of expert k to the uncertainty in the fuzzy forecast; $o_k\ge 0$. Chen and Wang [9] suggested that o_k should be between 1 and 4. s_k is the satisfaction level required by expert k; $0\le s_k\le 1$; m_k reflects the importance of the outlier to expert k; $m_k\ge 0$. Chen and Wang [9] suggested that m_k should be between 1 and 4. d_k is the required range by expert k, and $d_k\ge 0$. Maximizing Z_2 is equivalent to

$$\text{Max } Z_2'=\sum_{t=1}^{T}{s_t}^{m_k}.\tag{9}$$

However, there are two problems with Chen and Lin's method [8].

(1) The range of o_k was divided into six segments of the same width, and seven values of o_k, 1, 1.5, 2, 2.5, 3, 3.5, and 4, corresponding to seven linguistic terms (extremely insensitive, very insensitive, somewhat insensitive, moderate, somewhat sensitive, very sensitive, extremely sensitive) were provided for each expert to choose from. However, the effects of such a uniform partitioning were not proportionally reflected on the objective function. For example, the distance between 3.5 and 4 is the same as that between 1 and 1.5. If $\ln Y_{t3}-\ln Y_{t1}=0.5$, then $(\ln Y_{t3}-\ln Y_{t1})^4-(\ln Y_{t3}-\ln Y_{t1})^{3.5}=-0.026$ but $(\ln Y_{t3}-\ln Y_{t1})^{1.5}-(\ln Y_{t3}-\ln Y_{t1})^{1}=-0.146$. Obviously, $-0.026\gg -0.146$.

(2) The nonlinear objective functions and constraints are difficult to handle.

First, the range of o_k should be partitioned in such a way that the range of $(\ln Y_{t3}-\ln Y_{t1})^{o_k}$ can be divided into six equal parts. According to (7), $\ln Y_{t3}-\ln Y_{t1}$ is to be smaller than d_k. Therefore, the problem becomes dividing the interval $[d_k^1\ \ d_k^4]$ into six equal parts indicated with $[o_k(j)\ \ o_k(j+1)]$, $j=1\sim 6$; $o_k(1)=1$, $o_k(7)=4$. To this end, the following equations are to be solved individually:

$$\begin{aligned}
d_k^{o_k(2)} &= \frac{5d_k+d_k^4}{6}\longrightarrow o_k(2)=\ln\frac{\left(\left(5d_k+d_k^4\right)/6\right)}{\ln\left(d_k\right)},\\
d_k^{o_k(3)} &= \frac{4d_k+2d_k^4}{6}\longrightarrow o_k(3)=\ln\frac{\left(\left(4d_k+2d_k^4\right)/6\right)}{\ln\left(d_k\right)},\\
d_k^{o_k(4)} &= \frac{3d_k+3d_k^4}{6}\longrightarrow o_k(4)=\ln\frac{\left(\left(3d_k+3d_k^4\right)/6\right)}{\ln\left(d_k\right)},\\
d_k^{o_k(5)} &= \frac{4d_k+2d_k^4}{6}\longrightarrow o_k(5)=\ln\frac{\left(\left(2d_k+4d_k^4\right)/6\right)}{\ln\left(d_k\right)},\\
d_k^{o_k(6)} &= \frac{d_k+5d_k^4}{6}\longrightarrow o_k(6)=\ln\frac{\left(\left(d_k+5d_k^4\right)/6\right)}{\ln\left(d_k\right)}.
\end{aligned}\tag{10}$$

For example, if $d_k=0.5$, then $o_k(1)=1$, $o_k(2)=1.22$, $o_k(3)=1.5$, $o_k(4)=1.83$, $o_k(5)=2.26$, $o_k(6)=2.88$, and $o_k(7)=4$.

Subsequently, the nonlinear objective function and constraints can be converted into quadratic ones by minimizing the mean absolute error of approximation when $0\le x\le 1.2$ as

$$\text{Min}\sum_{x=0,0.01,\ldots}^{1.2}\frac{\left|x^h-\left(\alpha x^2+\beta x+\gamma\right)\right|}{121}.\tag{11}$$

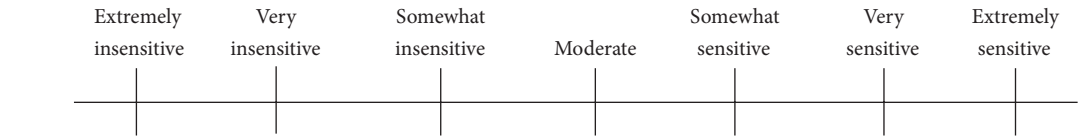
The range $[0\ \ 1.2]$ is wide enough for most possible yield values to fall into. To this end, a look-up table has been established after extensive numerical simulation; see Table 2. The mean absolute percentage error (MAPE) was less than 5%. The MAPE will not be a serious problem if it is easier to find the global optimal solution after approximation.

For some frequently used values especially, the fitted polynomials are shown below:

$$\begin{aligned}
x^{1.5} &\cong 0.5027x^2+0.5308x-0.0347,\\
x^{2.5} &\cong 1.4357x^2-0.4847x+0.0528,\\
x^{3} &\cong 1.8000x^2-0.8953x+0.1051,\\
x^{3.5} &\cong 2.0997x^2-1.2322x+0.1508,\\
x^{4} &\cong 2.3440x^2-1.5053x+0.1890.
\end{aligned}\tag{12}$$

The application method of Table 2 is shown in Figure 2.

After approximation, the following QP problems are solved instead.



NLP scale	1	1.5	2	2.5	3	3.5	4
QP scale	0	0.5027	1	1.4357	1.8	2.0997	2.344
	1	0.5308	0	−0.4847	−0.8953	−1.2322	−1.5053
	0	−0.0347	0	0.0528	0.1051	0.1508	0.189

Figure 2: The application of the look-up table.

Table 2: The look-up table.

h	α	β	γ
1.00	0	0	1
1.01	0.0093	0.9927	−0.0020
1.02	0.0187	0.9852	−0.0039
⋮	⋮	⋮	⋮
4.00	2.3440	−1.5053	0.1890

QP problem I is

$$\begin{aligned}\text{Min } Z_3 = \sum_{t=1}^{T} \Big(& \alpha(o(k)) \left(\ln Y_{t3} - \ln Y_{t1}\right)^2 \\ & + \beta(o(k)) \left(\ln Y_{t3} - \ln Y_{t1}\right) + \gamma(o(k)) \Big),\end{aligned} \tag{13}$$

subject to

$$\begin{aligned}\ln y_t &\geq \ln Y_{t1} + s_k \left(\ln Y_{t2} - \ln Y_{t1}\right), \\ \ln y_t &\leq \ln Y_{t3} + s_k \left(\ln Y_{t2} - \ln Y_{t3}\right), \\ \ln Y_{t1} &= a_1 - \frac{b_3}{t}, \\ \ln Y_{t2} &= a_2 - \frac{b_2}{t}, \\ \ln Y_{t3} &= a_3 - \frac{b_1}{t}, \\ 0 &\leq a_1 \leq a_2 \leq a_3, \\ 0 &\leq b_1 \leq b_2 \leq b_3, \\ t &= 1 \sim T.\end{aligned} \tag{14}$$

QP problem II is

$$\text{Max } Z_4 = \sum_{t=1}^{T} \left(\alpha(m(k)) s_t^{\,2} + \beta(m(k)) s_t + \gamma(m(k))\right) \tag{15}$$

subject to

$$\begin{aligned}\sum_{t=1}^{T} \Big(& \alpha(o(k)) \left(\ln Y_{t3} - \ln Y_{t1}\right)^2 \\ & + \beta(o(k)) \left(\ln Y_{t3} - \ln Y_{t1}\right) + \gamma(o(k)) \Big) \\ & \leq T \cdot d_k^{\,o_k}, \\ \ln y_t &\geq \ln Y_{t1} + s_t \left(\ln Y_{t2} - \ln Y_{t1}\right), \\ \ln y_t &\leq \ln Y_{t3} + s_t \left(\ln Y_{t2} - \ln Y_{t3}\right), \\ \ln Y_{t1} &= a_1 - \frac{b_3}{t}, \\ \ln Y_{t2} &= a_2 - \frac{b_2}{t}, \\ \ln Y_{t3} &= a_3 - \frac{b_1}{t}, \\ 0 &\leq a_1 \leq a_2 \leq a_3, \\ 0 &\leq b_1 \leq b_2 \leq b_3, \\ t &= 1 \sim T.\end{aligned} \tag{16}$$

Take the case in Chen and Wang [9] as an example; see Table 3. In NLP models, the four parameters o_k, s_k, m_k, and d_k were set to 1.1, 0.4, 1, and 0.71 in their study. The NLP models were converted into the QP ones. Then both methods were applied to predict the yield. After defuzzifying the yield forecasts with the centroid-of-gravity formula, the forecasting performances by the two methods were compared in Table 4.

(1) Obviously, the forecasting accuracy of the QP models was better. Considering MAPE, the proposed QP-based method reduced the prediction error up to 10%.

(2) On the other hand, it is also possible to use the proposed QP-based method to improve the prediction precision measured in terms of the average range of forecasts.

(3) In theory, as long as quadratic polynomials can very precisely fit the original nonlinear objective functions and constraints, the prediction accuracy

Table 3: The case in Chen and Wang [9].

t	1	2	3	4	5	6	7
Y_t	37.30%	58.50%	54.10%	74.10%	61.70%	80.00%	71.20%

Table 4: The performances by the two methods.

Model	The average range (precision)	Mean absolute percentage error (accuracy)
NLP I	0.34	14.5%
NLP II	0.37	17.2%
QP I	0.40	14.1%
QP II	0.28	14.3%

of the proposed QP-based method must be better than the original NLP-based method. Even with little error, a similar effect still exists. For this reason, it seems that it is not necessary to compare the subsequent collaboration step. Fuzzy collaborative forecasting methods based on improved individual forecasts should perform better.

(4) It is also easy to prove that the quadratic objective function and constraints are convex, because $\ln Y_{t3} \geq \ln Y_{t1}$ and $\alpha(o(k)) \geq 0$. That is very important to the global optimality of the optimal solution.

4. Discussion and Conclusion

Fuzzy collaborative forecasting methods have considerable potential for the semiconductor yield forecasting. In this field, the existing methods are mostly based on NLP models. Such a practice has some drawbacks. To solve these problems, we proposed a QP-based fuzzy collaborative forecasting method and made a fair comparison with instances from the literature.

According to the experimental results, the QP-based approach achieved a better prediction performance than the NLP-based method, because it is easier to find the global optimal solution in the QP-based method than the method based on NLP. In addition, the disagreements between the views of different experts are also better distinguished in the QP-based approach. Furthermore, the NLP models are in fact formulated subjectively and certainly can be replaced by any other formulation, to achieve a better performance. The error of estimating the NLP model with the QP models becomes unimportant as well. In other words, the future focus should be on the development of better QP models, rather than the approximation of the NLP models.

Acknowledgment

This work was supported by the National Science Council of Taiwan.

References

[1] T. Chen, "A FNP approach for evaluating and enhancing the long-term competitiveness of a semiconductor fabrication factory through yield learning modeling," *International Journal of Advanced Manufacturing Technology*, vol. 40, no. 9-10, pp. 993–1003, 2009.

[2] T. Chen, "A fuzzy mid-term single-fab production planning model," *Journal of Intelligent Manufacturing*, vol. 14, no. 3-4, pp. 273–285, 2003.

[3] T. Chen and M.-J. J. Wang, "Forecasting methods using fuzzy concepts," *Fuzzy Sets and Systems*, vol. 105, no. 3, pp. 339–352, 1999.

[4] A. M. Spence, "The learning curve and competition," *Bell Journal of Economics*, vol. 12, pp. 49–70, 1981.

[5] S. Majd and R. S. Pindyck, "The learning curve and optimal production under uncertainty," *Rand Journal of Economics*, vol. 20, no. 3, pp. 331–343, 1989.

[6] J. B. Mazzola and K. F. McCardle, "A Bayesian approach to managing learning-curve uncertainty," *Management Science*, vol. 42, no. 5, pp. 680–692, 1996.

[7] K. Anderson, "Innovative yield modeling using statistics," in *Proceedings of the 17th Annual SEMI/IEEE Advanced Semiconductor Manufacturing Conference (ASMC '06)*, pp. 219–221, May 2006.

[8] T. Chen and Y. C. Lin, "A fuzzy-neural system incorporating unequally important expert opinions for semiconductor yield forecasting," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 16, no. 1, pp. 35–58, 2008.

[9] T. Chen and Y. C. Wang, "An agent-based fuzzy collaborative intelligence approach for precise and accurate semiconductor yield forecasting," *IEEE Transactions on Fuzzy Systems*, 2013.

[10] J. Watada, H. Tanaka, and T. Shimomura, "Identification of learning curve based on possibilistic concepts," in *Applications of Fuzzy Set Theory in Human Factors*, Elsevier, The Netherlands, 1986.

[11] J. S. Lin, "Constructing a yield model for integrated circuits based on a novel fuzzy variable of clustered defect pattern," *Expert Systems With Applications*, vol. 39, no. 3, pp. 2856–2864, 2012.

[12] L. Wu, J. Zhang, and G. Zhang, "A fuzzy neural network approach for die yield prediction of wafer fabrication line," in *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09)*, pp. 198–202, August 2009.

[13] Y.-M. Chiang and H.-H. Hsieh, "Applying grey relational analysis and fuzzy neural network to improve the yield of thin-film sputtering process in color filter manufacturing," in *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM '09)*, pp. 408–413, June 2009.

[14] W. Pedrycz, "Collaborative architectures of fuzzy modeling," in *Computational Intelligence: Research Frontiers*, vol. 5050 of *Lecture Notes in Computer Science*, pp. 117–139, Springer, Berlin, Germany, 2008.

[15] T. Chen and Y. C. Wang, "A fuzzy-neural approach for global CO_2 concentration forecasting," *Intelligent Data Analysis*, vol. 15, no. 5, pp. 763–777, 2011.

[16] T. Chen, "A fuzzy-neural knowledge-based system for job completion time prediction and internal due date assignment in a wafer fabrication plant," *International Journal of Systems Science*, vol. 40, no. 8, pp. 889–902, 2009.

[17] O. Shai and Y. Reich, "Infused design. I. Theory," *Research in Engineering Design*, vol. 15, no. 2, pp. 93–107, 2004.

[18] O. Shai and Y. Reich, "Infused design. II. Practice," *Research in Engineering Design*, vol. 15, no. 2, pp. 108–121, 2004.

[19] E. Ostrosi, L. Haxhiaj, and S. Fukuda, "Fuzzy modelling of consensus during design conflict resolution," *Research in Engineering Design*, pp. 1–18, 2011.

[20] T. S. L. Lo, L. H. S. Luong, and R. M. Marian, "Holistic and collaborative demand forecasting process," in *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN '06)*, pp. 782–787, August 2006.

[21] G. Büyüközkan and Z. Vardaloğlu, "Analyzing of collaborative planning, forecasting and replenishment approachusing fuzzy cognitive map," in *Proceedings of the International Conference on Computers and Industrial Engineering (CIE '09)*, pp. 1751–1756, July 2009.

[22] G. B. Büyüközkan, O. Feyzioglu, and Z. Vardaloglu, "Analyzing CPFR supporting factors with fuzzy cognitive map approach," *World Academy of Science, Engineering and Technology*, vol. 31, pp. 412–417, 2009.

[23] N. Cheikhrouhou, F. Marmier, O. Ayadi, and P. Wieser, "A collaborative demand forecasting process with event-based fuzzy judgements," *Computers and Industrial Engineering*, vol. 61, no. 2, pp. 409–421, 2011.

[24] R. Poler, J. E. Hernandez, J. Mula, and F. C. Lario, "Collaborative forecasting in networked manufacturing enterprises," *Journal of Manufacturing Technology Management*, vol. 19, no. 4, pp. 514–528, 2008.

[25] T. Chen, Y.-C. Wang, and H. C. Wu, "A fuzzy-neural approach for remaining cycle time estimation in a semiconductor manufacturing factory—a simulation study," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2125–2139, 2009.

[26] T. Chen, "A hybrid look-ahead SOM-FBPN and FIR system for wafer-lot-output time prediction and achievability evaluation," *International Journal of Advanced Manufacturing Technology*, vol. 35, no. 5-6, pp. 575–586, 2007.

[27] W. Pedrycz and P. Rai, "A multifaceted perspective at data analysis: a study in collaborative intelligent agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 38, no. 4, pp. 1062–1072, 2008.

[28] T. Chen, "A hybrid fuzzy and neural approach with virtual experts and partial consensus for DRAM price forecasting," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1B, pp. 583–598, 2012.

[29] H. Gruber, *Learning and Strategic Product Innovation: Theory and Evidence for the Semiconductor Industry*, Elsevier, The Netherlands, 1994.