Rule-Based Fuzzy-Neural Networks Using the Identification Algorithm of the GA Hybrid Scheme

Ho-Sung Park and Sung-Kwun Oh

Abstract: This paper introduces an identification method for nonlinear models in the form of rule-based Fuzzy-Neural Networks (FNN). In this study, the development of the rule-based fuzzy neural networks focuses on the technologies of Computational Intelligence (CI), namely fuzzy sets, neural networks, and genetic algorithms. The FNN modeling and identification environment realizes parameter identification through synergistic usage of clustering techniques, genetic optimization and a complex search method. We use a HCM (Hard C-Means) clustering algorithm to determine initial apexes of the membership functions of the information granules used in this fuzzy model. The parameters such as apexes of membership functions, learning rates, and momentum coefficients are then adjusted using the identification algorithm of a GA hybrid scheme. The proposed GA hybrid scheme effectively combines the GA with the improved complex method to guarantee both global optimization and local convergence. An aggregate objective function (performance index) with a weighting factor is introduced to achieve a sound balance between approximation and generalization of the model. According to the selection and adjustment of the weighting factor of this objective function, we reveal how to design a model having sound approximation and generalization abilities. The proposed model is experimented with using several time series data (gas furnace, sewage treatment process, and NOx emission process data from gas turbine power plants).

Keywords: Fuzzy-neural networks, computational intelligence (CI), clustering, GA hybrid scheme, genetic algorithm, improved complex method.

1. INTRODUCTION

During the past few years, fuzzy neural networks (FNN) have emerged as one of the most active and fruitful areas of research in fuzzy logic and neural networks. It is concerned with the integration of these two fields in which significant advances have been made during the last decade [1,2]. There have been many attempts to synthesize FNN.

Fuzzy set theory has been introduced [3,4] to model uncertain and/or ambiguous characteristics inherent to experimental data. Since its inception, the research of fuzzy logic has been a focal point of various endeavors and has demonstrated many fruitful results both in theory and application [5]. In the early approaches, the generation of the fuzzy rules and the adjustment of its membership functions were based on *trial and error* and/or operator's experience. Subsequently, the designers have found it difficult to develop adequate fuzzy rules and membership functions to reflect the essence of the data. Moreover, a variety of information becomes lost or ignored on purpose when human operators articulate their experience in the form of linguistic rules. A collection of manually developed fuzzy rules is usually suboptimal. As a consequence, there is a genuine need for an optimization environment to construct and/or adjust a collection of linguistic rules. While there has been an impressive panoply of neuro-fuzzy approaches, a comprehensive solution is still to be developed. In this synergistic arrangement, the disadvantages of these two technologies tend to be compensated for when used in the context of fuzzy rule-based models.

In this paper, we use a linear fuzzy inference-based FNN model as the fuzzy set-based approach by the fuzzy partitions formed for the individual. This model has a faster learning speed and better convergence characteristics than other counterpart models. The structure of the network is constructed by partitioning fuzzy input-output space for each input variable. However this approach does exhibit certain limitations. The primary drawback comes with an inability to capture the essential characteristics of input-output

Manuscript received July 11, 2002; accepted November 21, 2002. This study was supported by the Korea Energy Management Corporation (2001).

Ho-Sung Park is with the Department of Control & Instrumentation Engineering, Wonkwang University, 570-749, Korea. (e-mail: neuron @wonkwang. ac.kr).

Sung-Kwun Oh is with the School of Electrical Electronic & Information Engineering, Workwang University, 570-749, Korea. (e-mail: ohsk@wonkwang.ac.kr).

data. To overcome such a dilemma, we use a HCM (Hard C-Means) clustering algorithm. As it concentrates on revealing the essential structure of the data, it helps build a blueprint of the network (more specifically the method helps determine initial apexes of the membership functions used in the model). In the sequel, using the identification algorithm of a GA hybrid scheme, we further optimize the proposed model. The hybrid identification algorithm dwells on the ideas of genetic algorithms (GAs) [6-8] as well as the improved complex algorithm [9]. In addition, we introduce an aggregate objective function [9] that takes into account both training data and testing data. This index aims at achieving a sound balance between approximation and prediction capabilities of the proposed model. Experimentally, the proposed model is discussed for time-series data for the gas furnace process [10], activated sludge process in sewage treatment systems [9] and NOx emission process data of gas turbine power plants [11].

2. RULE-BASED FUZZY-NEURAL NETWORKS

The structure of FNN emerges at a junction of fuzzy sets and neural networks. In this section, we discuss the type of "if-then" rules along with their development mechanisms, that is, the linear fuzzy inference-based FNN as rule base type. By means of the division of fuzzy input space, the model uses the FNN whose faster learning speed and better convergence characteristics than other FNN models is shown in Fig. 1.

The notation used here requires some clarification. The "boxes" and "circles" denote units of the FNN while "N" identifies a normalization procedure applied to the membership grades of the input variable x_i. The output $f_i(x_i)$ of the "Σ" neuron is described by some nonlinear function f_i. f_i is not necessarily a sigmoid function encountered conventional neural network but we allow for more flexibility in this regard. Finally, the output of the FNN $\hat{y}$ is governed by the following expression,

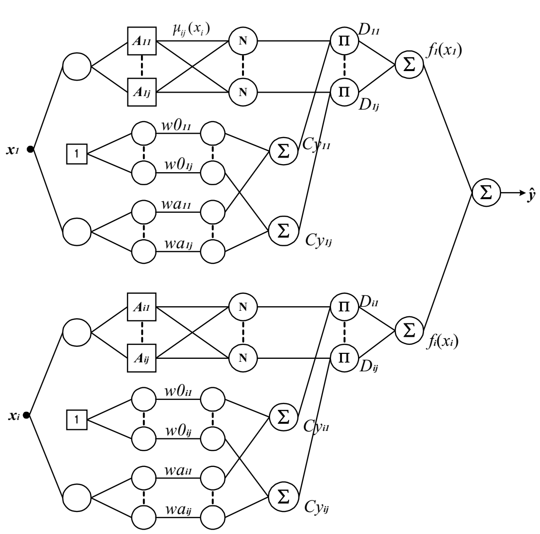


Fig. 1. Linear fuzzy inference-based FNN structure.

$$\hat{y} = f_1(x_1) + f_2(x_2) + \cdots f_m(x_m) = \sum_{i=1}^{m} f_i(x_i). \tag{1}$$

As previously mentioned, FNN is implied by the introduced fuzzy partition of each input variable. In this sense, we can regard each f_i given by (1) as the following rules,

$$R^j : IF \ \ x_i \ \ is \ \ A_{ij}, \ \ then \ \ Cy_{ij} = w0_{ij} + x_i wa_{ij}. \tag{2}$$

To be more specific, R^j is the *j-th* fuzzy rule while A_{ij} denotes a fuzzy variable of the premise of the corresponding fuzzy rule and represents membership function μ_{ij}. $w0_{ij}$ is a constant consequence of the rule and wa_{ij} is an input variable consequence of the rule. Each membership function in the premise part of fuzzy rule is assigned to be complementary with neighboring one.

The inference result coming from (2) follows a standard center of gravity aggregation.

$$f_i(x_i) = \sum_{j=1}^{n} \mu_{ij}(x_i) \cdot (w0_{ij} + x_i wa_{ij}) \Big/ \sum_{j=1}^{n} \mu_{ij}(x_i). \tag{3}$$

The learning of FNN is realized by adjusting connections of the neurons and as such it follows a standard Back-Propagation (BP) algorithm, cf [12-13]. In this study, we use the Euclidean error as a performance measure,

$$E_p = (y_p - \hat{y}_p)^2 \tag{4}$$

where E_p is an error measure for the *p-th* data, y_p is the *p-th* target output data and $\hat{y}_p$ stands for the *p-th* actual output of the model for this specific data point. For n input-output data pairs, an overall performance index arises as a sum of the errors,

$$E = \frac{1}{n} \sum_{p=1}^{n} (y_p - \hat{y}_p)^2. \tag{5}$$

Here, we look at the updates of the values of wa_{ij}. As far as learning is concerned, the connections change in a standard fashion,

$$wa(new) = wa(old) + \Delta wa \tag{6}$$

where the updated formula follows the gradient descent method, namely,

$$\Delta wa_{ij} = \eta_a \left(-\frac{\partial E_p}{\partial wa_{ij}} \right) \tag{7}$$

where η_a is a positive learning rate.

Moreover, we have

$$-\frac{\partial E_p}{\partial wa_{ij}}=-\frac{\partial E_p}{\partial \hat{y}_p}\cdot\frac{\partial \hat{y}_p}{\partial f_i(x_i)}\cdot\frac{\partial f_i(x_i)}{\partial D_{ij}}\cdot\frac{\partial D_{ij}}{\partial Cy_{ij}}\cdot\frac{\partial Cy_{ij}}{\partial wa_{ij}}. \quad (8)$$

Each part from the right side in (8) is expressed in the form,

$$-\frac{\partial E_p}{\partial \hat{y}_p}=-\frac{\partial}{\partial \hat{y}_p}(y_p-\hat{y}_p)^2=2(y_p-\hat{y}_p),\quad \frac{\partial \hat{y}_p}{\partial f_i(x_i)}=1,$$

$$\frac{\partial f_i(x_i)}{\partial D_{ij}}=1,\quad \frac{\partial D_{ij}}{\partial Cy_{ij}}=\mu_{ij}(x_i),\quad \frac{\partial Cy_{ij}}{\partial wa_{ij}}=x_i. \quad (9)$$

Finally, we obtain,

$$\Delta wa_{ij} = 2\cdot\eta_a\cdot(y_p - \hat{y}_p)\cdot\mu_{ij}(x_i)\cdot x_i. \quad (10)$$

Quite commonly to accelerate convergence, a momentum term is added to the learning expression. The momentum itself is defined in the form,

$$m(t)=w(t)-w(t-1) \quad (11)$$

The complete update formula combining the previously discussed components reads as,

$$\begin{aligned}\Delta wa_{ij} &= 2\cdot\eta_a\cdot(y_p - \hat{y}_p)\cdot\mu_{ij}(x_i)\cdot x_i \\ &+ \alpha_a(wa_{ij}(t) - wa_{ij}(t-1))\end{aligned} \quad (12)$$

Here the momentum coefficient, α_a, is constrained to the unit interval.

3. OPTIMIZATION OF FNN BY THE GA HYBRID SCHEME

The task of optimizing a complex system comprises at least two challenges for the system designer. First, a class of optimization algorithms must be chosen that is applicable to the system. Second, various parameters of the optimization algorithm need to be modified.

There is no guarantee that traditional GAs will give an optimal solution or arrangement, only that the solution will be near-optimal in the light of the specific fitness function used in the evaluation of the many possible solutions generated. The Complex Method is based on a sequential direct search technique, and no derivatives are required. However, this encounters the difficult problem surrounding selection of initial value. Therefore, if an incorrect initial value is selected, it may not converge to the local minimum point. In this study, the hybrid identification algorithm for dynamic parameters of fuzzy neural networks is sought, which combines the abilities of GAs and the improved complex method, thus resulting in enhanced performance. An overall flowchart of the design process indicating clearly how optimization mechanisms of the FNN model are employed is visualized in Fig. 2.

3.1. HCM clustering method

It is worth emphasizing that the HCM clustering method has been used extensively not only to organize and categorize data, but it also becomes significant in data compression and model identification. For the sake of completeness, let us recall the essence of the HCM algorithm.

Suppose that we are given a set of data X={$\mathbf{x}_1$, $\mathbf{x}_2$, ⋯ , $\mathbf{x}_n$}, where $\mathbf{x}_k$ =[x_{k1},⋯ , x_{km}], n is the number of data and m is the number of variables. Let P(X) be the power set of X, that is, the set of all the subsets of X. A hard c-partition of X is the family $\{A_i \in P(X): 1\le i\le c\}$ such that $\bigcup_{i=1}^{c} A_i = X$ and $A_i \cap A_j=\phi$ for $1\le i\ne j\le c$. Here c indicates the number of clusters. Each A_i is viewed as a cluster, so {A_1,⋯ , A_c} partitions X into c clusters. The hard c-partition can be reformulated through the characteristic (membership) function of the element $\mathbf{x}_k$ in A_i. Specifically, define

$$u_{ik}=\begin{cases}1, & \mathbf{x}_k\in A_i\\ 0, & \mathbf{x}_k\notin A_i\end{cases} \quad (13)$$

where $\mathbf{x}_k \in X$, $A_i \in P(X)$ and i=1,2,⋯ ,n. Clearly, u_{ik} = 1 signifies that $\mathbf{x}_k$ belongs to cluster A_i. Given the value of u_{ik}, we can uniquely determine a hard c-partition of X, and vice versa. The elements of the partition matrix u_{ik} satisfy the following three conditions,

$$u_{ik}\in\{0,1\},\ 1\le i\le c,\ 1\le k\le n \quad (14)$$

$$\sum_{i=1}^{c} u_{ik}=1,\ \forall k\in\{1,2,\cdots,n\} \quad (15)$$

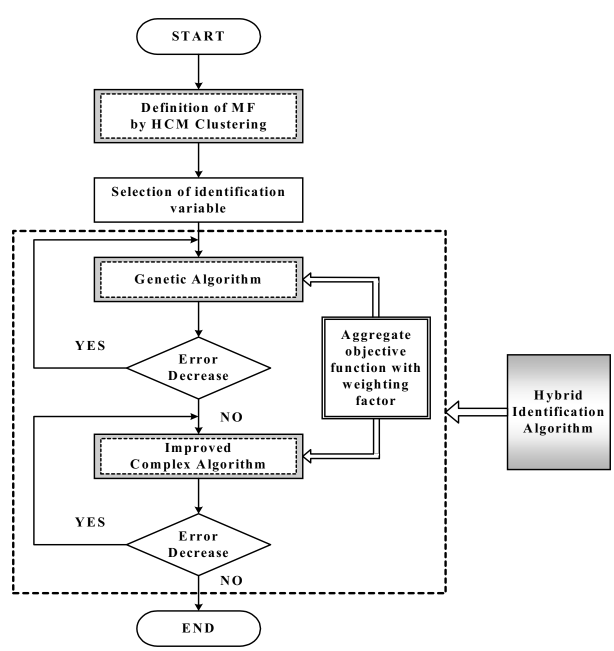


Fig. 2. An overall flowchart of the design of the FNN model.

$$0 < \sum_{k=1}^{n} u_{ik} < n, \quad \forall i \in \{1, 2, \cdots, c\}. \tag{16}$$

At the interpretation end, (14) and (15) represent that each $\mathbf{x}_k \in X$ should belong to one and only one cluster. (16) requires that each cluster A_i must contain at least one and at most n-1 data point(s). Collecting u_{ik} with $1 \square i \square c$ and $1 \square k \square n$ into a c×n matrix $\mathbf{U}=[u_{ik}]$ allowed us to obtain the matrix representation for hard c-partitions, defined as follows.

$$M_C = \left\{ \mathbf{U} \mid u_{ik} \in \{0,1\},\ \sum_{i=1}^{c} u_{ik} = 1,\ 0 < \sum_{k=1}^{n} u_{ik} < n \right\}. \tag{17}$$

Step 1: Determine the number of clusters $c (2 \le c < n)$ and initialize the partition matrix $\mathbf{U}^{(0)} \in M_C$

Step 2: Calculate the center vectors $\mathbf{v}_i$ of each cluster:

$$\mathbf{v}_i^{(r)} = \{v_{i1}, v_{i2}, \cdots, v_{ij}, \cdots, v_{im}\} \tag{18}$$

$$v_{ij}^{(r)} = \sum_{k=1}^{n} u_{ik}^{(r)} \cdot x_{kj} \Bigg/ \sum_{k=1}^{n} u_{ik}^{(r)} \tag{19}$$

where, $[u_{ik}]= \mathbf{U}^{(r)}$, $i = 1, 2, \cdots, c$, $j=1, 2, \cdots, m$.

Step 3: Update the partition matrix $\mathbf{U}^{(r)}$; these modifications are based on the standard Euclidean distance function between the data points and the prototypes,

$$d_{ik} = d(\mathbf{x}_k - \mathbf{v}_i) = \|\mathbf{x}_k - \mathbf{v}_i\| = \left[\sum_{j=1}^{m} (x_{kj} - v_{ij})^2 \right]^{1/2}, \tag{20}$$

$$u_{ik}^{(r+1)} = \begin{cases} 1 & d_{ik}^{(r)} = \min\{d_{jk}^{(r)}\} \text{ for all } j \in c \\ 0 & \text{otherwise} \end{cases}. \tag{21}$$

Step 4: Confirm termination criterion. If

$$\| \mathbf{U}^{(r+1)} - \mathbf{U}^{(r)} \| \le \varepsilon \text{ (tolerance level)} \tag{22}$$

stop; otherwise set $r = r + 1$ and return to step 2.

3.2. Genetic algorithms

Genetic algorithms [6-8] have proven to be useful in the optimization of such problems because of their ability to efficiently use historical information to obtain new solutions with enhanced performance and the global nature of search that is supported there. Genetic algorithms are also theoretically and empirically proven to support robust searches in complex search spaces. Moreover, they do not become trapped in local minima as opposed to gradient decent techniques that are quite susceptible to this shortcoming. GAs are population-based optimization techniques.

In this paper, for the optimization of the FNN model, GAs use the binary type serial method, roulette-wheel type in the selection operator, one-point crossover type in the crossover operator, and invert in the mutation operator. Here, we use 100 generations, 60 populations, 10 bits per string, a crossover rate equal to 0.6, and mutation probability equal to 0.1. A chromosome used in the genetic optimization consists of a string including vertical point of membership functions for each input variable, learning rate, and momentum coefficient.

3.3. Improved complex algorithms

Usually, by combining these optimization tasks we end up with a problem that is highly nonlinear and may not fit well into the domain of gradient-based techniques. To alleviate the problem, we propose to use an auto-tuning algorithm that is an adaptation of the improved complex algorithm [9].

We establish the algorithm by augmenting the method of a simplex concept to the complex method-constrained optimization technique. The proposed optimal auto-tuning algorithm known as the improved complex algorithm, is the constrained complex method of the form,

$$\text{Minimize } f(\boldsymbol{X}) \tag{23}$$

subject to

$$g_j(\boldsymbol{X}) \square 0, \qquad j = 1, 2, \cdots, m \tag{24}$$

$$\boldsymbol{X}_i^{(l)} \square \boldsymbol{X}_i \square \boldsymbol{X}_i^{(u)}, \qquad i = 1, 2, \cdots, n \tag{25}$$

where the superscripts l and u denote the lower and upper boundary of the corresponding variable.

In essence, it can be viewed as a sequence of six basic steps.

Step 1: The parameters to be optimized include elements of the FNN model. These consist of the apexes of membership function, learning rates, and momentum coefficients.

They are defined as $\boldsymbol{X}_k=(x_1^k, x_2^k, \cdots, x_n^k; k = 1, 2, \cdots, n, n+1, \cdots, m)$ and form the points in an "n" dimensional space. In general, the value of "m" is selected as being equal $2n$ (where, n is the number of the initial vertices).

Step 2: The initial values of a, $\square$ and $\square$ are specified using the Reflection, Expansion and Contraction of the simplex concept as follows,

I) Reflection : $\boldsymbol{X}_r = \boldsymbol{X}_o + a\ (\boldsymbol{X}_o - \boldsymbol{X}_h)$ (26)

II) Expansion : $\boldsymbol{X}_e = \boldsymbol{X}_o + \square(\boldsymbol{X}_r - \boldsymbol{X}_o)$ (27)

III) Contraction : $\boldsymbol{X}_c = \boldsymbol{X}_o + \square(\boldsymbol{X}_h - \boldsymbol{X}_o)$. (28)

Step 3: $\boldsymbol{X}_h$ and $\boldsymbol{X}_l$ are the vertices corresponding to the maximum function value $f(\boldsymbol{X}_h)$ and the minimum function value $f(\boldsymbol{X}_l)$. $\boldsymbol{X}_o$ is the centroid of all the points $\boldsymbol{X}_l$ except $i=h$. The reflection point $\boldsymbol{X}_r$ is given by (26), with $\boldsymbol{X}_h$ = max $f(\boldsymbol{X}_i)$, (i=1, 2, $\cdots$,k),

$$\mathbf{X}_o = \frac{(\sum_{i=1}^{n}\mathbf{X}_i - \mathbf{X}_h)}{m-1} \quad \text{and} \quad a = \frac{\|\mathbf{X}_r - \mathbf{X}_o\|}{\|\mathbf{X}_h - \mathbf{X}_o\|}.$$

If $\boldsymbol{X}_r$ does not satisfy the constraints, a new point $\boldsymbol{X}_r$ is generated by $\boldsymbol{X}_r = (\boldsymbol{X}_o + \boldsymbol{X}_r)/2$. This process is repeated until $\boldsymbol{X}_r$ does satisfy the constraints.

Step 4: If a reflection process gives a point $\boldsymbol{X}_r$ for which $f(\boldsymbol{X}_r) < f(\boldsymbol{X}_l)$, i.e. if the reflection produces a new minimum, we expand $\boldsymbol{X}_r$ to $\boldsymbol{X}_e$ by (27), with $\gamma = \frac{\|\mathbf{X}_e - \mathbf{X}_o\|}{\|\mathbf{X}_r - \mathbf{X}_o\|} > 1$.

If $\boldsymbol{X}_e$ does not satisfy the constraints, a new point $\boldsymbol{X}_e$ is generated by $\boldsymbol{X}_e = (\boldsymbol{X}_o + \boldsymbol{X}_e)/2$. This process is repeated until $\boldsymbol{X}_e$ does satisfy the constraints. If $f(\boldsymbol{X}_e) < f(\boldsymbol{X}_l)$, we replace the point $\boldsymbol{X}_h$ by $\boldsymbol{X}_e$ and restart the process of reflection. On the other hand, if $f(\boldsymbol{X}_e) > f(\boldsymbol{X}_l)$, we replace the point $\boldsymbol{X}_h$ by $\boldsymbol{X}_r$, and begin the reflection process again.

Step 5: If the reflection process produces a point $\boldsymbol{X}_r$ for which $f(\boldsymbol{X}_r) > f(\boldsymbol{X}_i)$, for all i except $i = h$. If $f(\boldsymbol{X}_r) < f(\boldsymbol{X}_h)$, then we replace the point $\boldsymbol{X}_h$ by $\boldsymbol{X}_r$. In this case, we contract the simplex as in (28), with $\beta = \frac{\|\mathbf{X}_c - \mathbf{X}_o\|}{\|\mathbf{X}_h - \mathbf{X}_o\|}$. If $f(\boldsymbol{X}_r) > f(\boldsymbol{X}_h)$, we use $\boldsymbol{X}_c$ without changing the previous point $\boldsymbol{X}_h$. If $\boldsymbol{X}_c$ does not satisfy the constraints, a new point $\boldsymbol{X}_c$ is generated with $\boldsymbol{X}_c = (\boldsymbol{X}_o + \boldsymbol{X}_c)/2$. This process is conducted repeatedly until $\boldsymbol{X}_c$ does satisfy the constraints. If the contraction process produces a point $\boldsymbol{X}_c$ for which $f(\boldsymbol{X}_c) < \min[f(\boldsymbol{X}_h), f(\boldsymbol{X}_r)]$, we replace the point $\boldsymbol{X}_h$ by $\boldsymbol{X}_c$. We then proceed with the reflection once again. On the other hand, if $f(\boldsymbol{X}_c)$ □ $\min[f(\boldsymbol{X}_h), f(\boldsymbol{X}_r)]$, we replace all $\boldsymbol{X}_i$ by $(\boldsymbol{X}_i + \boldsymbol{X}_l)/2$, and begin the reflection process again.

Step 6: This method is assumed to have converged whenever the standard deviation of the function at the vertices of the current simplex is smaller than some prescribed small quantity as follows:

$$Q = \left\{ \sum_{i=1}^{n+1} \frac{[f(\mathbf{X}_i) - f(\mathbf{X}_o)]^2}{n+1} \right\}^{1/2} \le \varepsilon . \tag{29}$$

If Q does not satisfy (29), we go to step 3.

In this study, the reflection, expansion, and contraction coefficients which are the initial parameters of the improved complex algorithm are set as $\alpha = 1$, $\gamma = 2$ and $\beta = 0.5$, respectively.

3.4. Identification algorithm by the GA hybrid scheme

GAs are global optimization techniques that avoid many shortcomings exhibited on conventional search techniques when completed in a large and in a complex space. However, GAs perform a blind search and do not guarantee local convergence. That is, GAs tend to efficiently explore various regions of the decision space with a high probability of finding improved solutions [6]. There is no guarantee that the final solution obtained using a GA is the global optimal solution to a problem.

The complex method is a mathematical programming technique that prescribes a systematic procedure for obtaining a local optimal solution to a nonlinear, constrained optimization problem. The trouble with this method centers on the selection of a starting point. To alleviate these difficulties, we've considered the hybrid identification algorithm. It combines the genetic algorithm effectively with the improved complex method to guarantee both global optimization and local convergence.

The hybrid identification algorithm assumes the advantage of GAs and the improved complex method, that is, the algorithm approaches a near-optimal solution and then rapidly reaches the global minimum. Therefore, the hybrid algorithm addresses the problems of the GAs that stay at a near-global minimum without reaching it and the improved complex method that exhibits difficulties in determining the initial points from which a global solution can be reached.

3.5. The objective function with weighting factor

We elaborate on the performance index. The objective function for the training data and testing data assumes the form

$$f(\mathrm{PI}, \mathrm{E\_PI}) = \theta \times \mathrm{PI} + (1-\theta) \times \mathrm{E\_PI}(\mathrm{V\_PI}) \tag{30}$$

and is utilized as a cost function of the FNN model.

θ and $(1-\theta)$ are two weighting factors for PI and E_PI, respectively. PI and E_PI (or V_PI) denote the values of the performance index for the training data and testing data (or validation data), respectively. For the purpose of minimizing this objective function, all parameters of the premise membership function of the triangular function are modified so as to be optimal.

Depending upon the values of the weighting factor, let us discuss several specific cases of the objective function.

1. If $\theta = 1$ then the model becomes optimized based on the training set. No testing set is taken into consideration.
2. If $\theta = 0.5$ then the training and testing sets are taken into account.
3. If $\theta = \alpha$ ($\alpha \in [0, 1]$) then the choice of α establishes a certain trade-off between the approximation and generalization aspects of the FNN model.
4. In general, θ can be selected and adjusted inde-

pendently.

4. EXPERIMENTAL STUDIES

Once the identification methodology has been established, one can proceed with intensive experimental studies. In this section, we provide three numerical examples to evaluate the advantages and the effectiveness of the proposed approach. These include gas furnace data [10], the sewage treatment process [9], and NOx emission process data of gas turbine power plants [11].

4.1. Gas furnace process

In this section, the proposed model is applied to the time-series data of a gas furnace utilized by Box and Jenkins[10]. We try to model the gas furnace using 296 pairs of input-output data. The flow rate of methane gas, $u_m(t)$ used in the laboratory changes from –2.5 to 2.5. The control $u(t)$ used in real process, ranges from 0.5 to 0.7 following the expression:

$$u(t) = 0.6 - 0.048 \times u_m(t) \quad (31)$$

u denotes the flow rate of methane as input, the output stands for the carbon dioxide density i.e., the outlet gas. In order to carry out the simulation, we use two-input data ($u(t-3)$,$y(t-1)$) and one-output data ($y(t)$). In addition, the total data set of the gas furnace process is partitioned into two parts, the training data and the testing data. Table 1 summarizes computational aspects of the approach as well as elaborates on the remaining parameters of the hybrid algorithm. Fig. 3 shows the membership functions of each input variable according to the partition of fuzzy input spaces by a Min-Max method and the HCM clustering method.

Table 1. Computational overhead and a list of parameters of the optimization method.

Genetic algorithms		Improved complex algorithm	
Generation	100	a	1
Population	60	□	0.5
String	10	□	2
Crossover rate	0.6	ε	1×10^{-6}
Mutation probability	0.1	Complex iterations	500
FNN iterations	300	FNN iterations	100

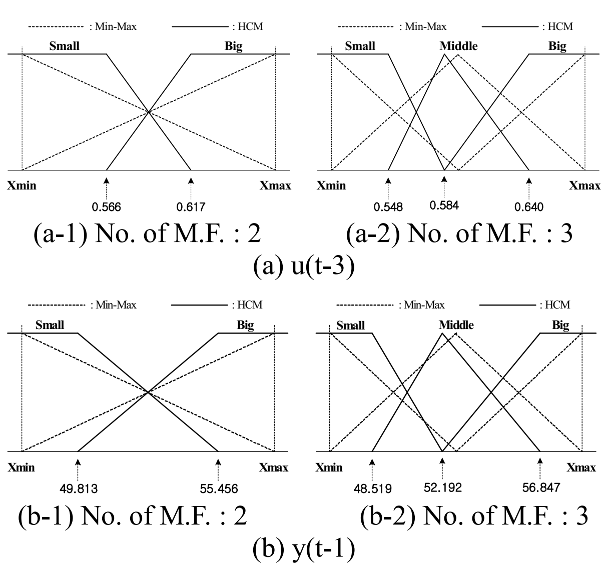


Fig. 3. Definition of initial membership functions of the gas furnace process by HCM.

Table 2. Performance index of gas furnace process according to a change in the number of membership functions.

θ	No. of membership functions					
	2 : 2		3 : 2		3 : 3	
	PI	E_PI	PI	E_PI	PI	E_PI
0.0	0.050	0.264	0.061	0.271	0.078	0.278
0.2	0.049	0.264	0.048	0.271	0.049	0.282
0.4	0.044	0.267	0.041	0.274	0.033	0.293
0.5	0.037	0.273	0.037	0.276	0.029	0.296
0.6	0.034	0.276	0.033	0.281	0.028	0.298
0.8	0.027	0.296	0.028	0.294	0.025	0.306
1.0	0.023	0.339	0.024	0.348	0.022	0.337

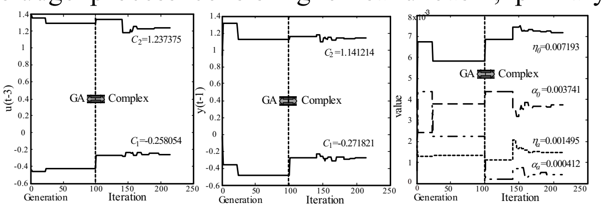
Table 2 shows the values of the performance index of the FNN obtained with the use of the hybrid identification algorithm according to the weighting factor and the number of membership functions of each input variable. The hybrid identification algorithm extracts the optimal parameters of FNN such as apexes of membership function, learning rate, and momentum coefficient. As illustrated in Table 2, according to the selection and adjustment of a weighting factor we can design the desired model that contains the intention of the designer considering approximation and generalization ability.

In case the weighting factor and the number of rules are chosen as 0.8 and 4 respectively, the parameters of the algorithm and the resulting membership functions are shown in Fig. 4. Moreover, in conjunction with Fig.4, the optimized parameters of the membership functions by GA and hybrid algorithm are shown in Fig. 5.

Fig. 6 illustrates the optimization process by visualizing the performance index in successive cycles (generation and iterations) of the hybrid algorithm. It also shows the preferred network architecture (the weighting factor θ is set to 0.8 in the FNN model). Table 3 provides a comparison of the proposed model with other models being already proposed in the literature. The comparison is realized on the basis of the same performance index for the training set and the testing set.

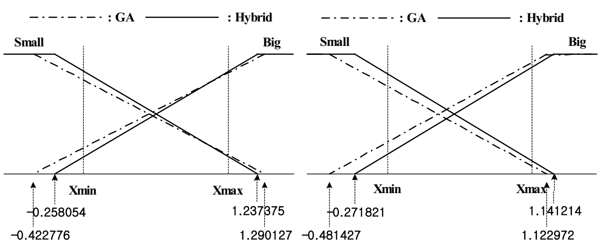
4.2. Sewage treatment process

Sewage treatment generally uses the activated sludge process consisting of sand basin, primary



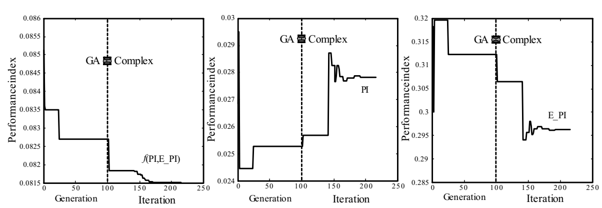
(a) u(t-3) (b) y(t-1) (c) η and α

Fig. 4. Convergence process of optimal parameters for the linear fuzzy inference-based FNN model by hybrid algorithm(θ = 0.8).



(a) u(t-3) (b) y(t-1)

Fig. 5. The final tuned values of membership functions by GA and hybrid algorithm(θ = 0.8).



(a-1) f(PI, E_PI) (a-2) PI (a-3) E_PI

Fig. 6. The optimization process of each performance index for FNN model by hybrid algorithm(No. of rules: 4, θ = 0.8).

Table 3. Comparative summary of existing fuzzy models.

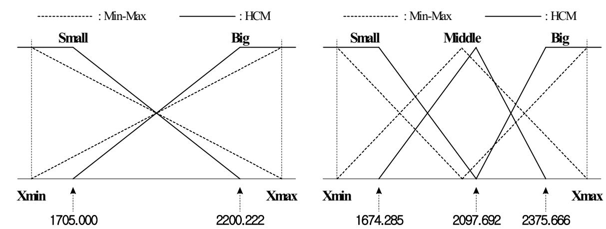
MODEL			PI	E_PI	No. of rules
Oh and Pedrycz's Model [9]	Simplified		0.024	0.328	4
	Linear		0.022	0.326	4
			0.021	0.364	6
			0.020	0.333	8
Kim, et al.'s model [14]			0.034	0.244	2
Lin, et al's model [15]			0.071	0.261	4
Our model	GA	θ=0.5	0.042	0.302	6
	Hybrid	θ=0.8	0.027	0.296	4
			0.028	0.294	5
		θ=0.5	0.029	0.296	6

sedimentation basin, aeration tank and final sedimentation basin [9]. In this experiment, we use a data set coming from the sewage treatment system plant in Seoul, Korea.

The proposed model is carried out using 52 pair of inputs-output data obtained from the activated sludge

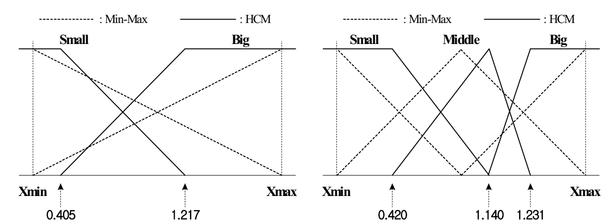
Table 4. Summary of the parameters of the optimization and computational effort.

Genetic algorithms		Improved complex algorithm	
Generation	100	a	1
Population	60	□	0.5
String	10	□	2
Crossover rate	0.6	ε	1× 10^{-6}
Mutation probability	0.1	Complex iterations	500
FNN iterations	500	FNN iterations	500



(a-1) No. of M.F. : 2 (a-2) No. of M.F. : 3

(a) MLSS



(b-1) No. of M.F. : 2 (b-2) No. of M.F. : 3

(b) WSR

Fig. 7. Definition of initial membership functions of sewage treatment process by HCM.

process. From four input variables (MLSS, WSR, RRSP, and DOSP), we choose two input variables (MLSS, WSR) that minimize the evaluation, and extract more than two fuzzy partitions (fuzzy sets LOW and HIGH) from each input-output pair of data. Table 4 shows computational costs as well as the related parameters used in the hybrid algorithm.

Fig. 7 shows the membership functions of two inputs variables according to the partition of fuzzy input spaces by a Min-Max method and the HCM clustering method. Table 5 shows the values of the performance index of the FNN obtained using the hybrid identification algorithm. The hybrid identification algorithm extracts the optimal parameters of FNN such as apexes of membership functions, learning rate, and momentum coefficients.

According to selection of the weighting factor θ and the number of membership functions of each input variable, the linear fuzzy inference-based FNN models are included in Table 5. When we select two input variables, MLSS and WSR, and set up θ to be equal to 0.5, the FNN model has the preferred network architecture, and the values of the indexes PI and E_PI are equal to 12.932 and 7.786, respectively.

Table 6 summarizes the results of a comparative analysis between the proposed model and other models. Again, the performance of the proposed model is far better both in the sense of its prediction (generalization) and approximation abilities. As shown in Table 6, from the viewpoint of PI as well as E_PI, the results of models optimized by hybrid algorithm are improved in comparison with other models and GA-based FNN model. In particular, Table 6 shows that in comparison with the PI, the E_PI (performance index for testing data) of the FNN model obtained using the hybrid identification algorithm is far better than in case when GA is employed. As we know from Table 6, we can obtain better performance through the hybrid identification algorithm in data of heavy nonlinearity than uniformly distributed gas furnace data.

4.3. NOx emission process data of gas turbine power plant

The NOx emission process is also modeled using the data from gas turbine power plants. The NOx emission process of a GE gas turbine power plant located in Virginia, U.S.A., is chosen as a model in this study. The input variables include AT (Ambient Temperature at site), CS (Compressor Speed), LPTS (Low Pressure Turbine Speed), CDP (Compressor Discharge Pressure), and TET (Turbine Exhaust Temperature). The output variable is NOx [11]. The performance index is defined by (5). We have taken into consideration 260 pairs of the original input-output data. 130 pairs out of 260 pairs of input-output data set are used as the training data set; the remaining part serves as the testing data set.

Using NOx emission process data, the regression equation is obtained as follows.

$$y = -163.77341 - 0.06709x_1 + 0.00322\,x_2 + 0.00235\,x_3 + 0.26365\,x_4 + 0.20893\,x_5\,. \quad (32)$$

This simple model comes with the value of PI=17.68 and E_PI=19.23. We will be using as a reference point when discussing FNN models. Table 7 shows computational cost and the related parameters used in the hybrid algorithm.

In the case of NOx emission process data, there are numerous input variables as well as a quantity of data. To look into the performance characters of this process, overall dataset pairs of I/O data are split into three parts, namely the training dataset, validation dataset and testing dataset. By using these dataset, we get the values of performance index according to the change in the number of membership functions for each input variable. Table 8 shows the performance index of the proposed FNN according to the number of membership functions.

Table 5. Performance index of sewage treatment process according to the change of number of membership functions (MLSS, WSR).

θ	No. of membership functions (Total no. of rules)					
	2 : 2 (4)		3 : 2 (5)		3 : 3 (6)	
	PI	E_PI	PI	E_PI	PI	E_PI
0.0	15.038	8.014	16.490	8.781	13.395	7.706
0.2	14.734	8.241	14.513	8.879	13.275	7.619
0.4	12.891	9.355	12.510	10.396	13.019	7.704
0.5	12.307	9.828	11.998	10.807	12.932	7.786
0.6	11.847	10.392	11.675	11.205	9.343	12.250
0.8	10.975	12.142	10.924	13.063	8.750	13.422
1.0	10.158	24.710	9.118	16.585	7.705	31.943

Table 6. Comparison of performance with other modeling methods (Input variables – MLSS, WSR).

MODEL			PI	E_PI	No. of rules
Oh and Pedrycz's Model [9]	Simplified		13.72	16.20	4
			14.10	16.56	6
			12.80	15.91	8
	Linear		6.39	54.23	4
			1.46	8.06e+4	6
			0.001	923.32	8
Fuzzy Model[16]	Simplified		12.35	11.17	6
	Linear		0.001	126.91	6
FNN Model [17]			13.75	9.24	4
			12.34	10.19	9
Our model	GA	θ=0.5	12.59	10.22	4
			10.96	12.72	6
	Hybrid	θ=0.5	12.30	9.82	4
			11.99	10.80	5
			12.93	7.78	6

Table 7. Parameters of the optimization environment and computational effort.

Genetic algorithms		Improved complex algorithm	
Generation	100	a	1
Population	60	□	0.5
String	10	□	2
Crossover rate	0.6	ε	1×10^{-6}
Mutation probability	0.1	Complex iterations	500
FNN iterations	1000	FNN iterations	500

Here, PIs, V_PIs and E_PIs denote the performance index for the training dataset, validation dataset and testing dataset, respectively. When the number of membership functions for each input variable increases, we obtain better performance of the model. As shown in Table 8, in case of using 6 MFs per input variable, the variation ratio (slope) of the perform

Table 8. Performance index according to the change of number of MFs.

Performance index / No. of MFs per input	Linear fuzzy inference-based FNNs		
	PIs	V_PIs	E_PIs
2	14.509	13.700	4.605
3	8.763	10.924	3.613
6	3.810	5.782	2.457

Table 9. Performance index as a function of the weighting factor.

θ	Linear fuzzy inference-based FNNs			
	GA		Hybrid (GA+Complex)	
	PI	E_PI	PI	E_PI
0.0	4.461	5.496	3.683	5.630
0.2			3.725	5.291
0.4			3.630	5.627
0.5	4.038	6.028	3.830	5.397
0.6			3.574	5.481
0.8			3.547	5.813
1.0	3.795	6.814	3.640	6.468

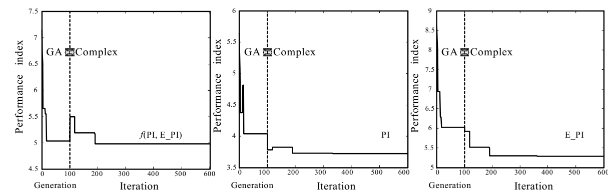
ance index of the FNN model does not change radically. E_PIs is particularly slow-moving. This led us to accept six fuzzy sets for each input variable.

Equally as other experiment studies, NOx emission data is split into two parts, namely training dataset (PI) and testing dataset (E_PI). And the number of membership functions for each input variable is set to six. Table 9 includes the values of the performance index of the FNN model derived when using the hybrid identification algorithm and the weighting factor is superior to the one we obtained through genetic optimization. For the linear fuzzy inference-based FNN, the results (that is PI=3.725, E_PI=5.291) are reported when using weighting factor (θ) equal to 0.2.

Refer to Table 9. From the results of this preferred network architecture, Fig. 8 depicts the optimization process by showing the values of the performance index in successive cycles of the hybrid algorithm.

5. CONCLUSIONS

In this paper, the efficient identification technique is presented that automatically extracts the optimal parameters of the linear fuzzy inference-based FNN using the identification algorithm of a GA hybrid scheme and the weighting factor of an objective function. The development of the rule-based fuzzy neural networks dwells on the technologies of Computational Intelligence (CI). The underlying idea deals with an optimization of information granules by exploiting techniques of clustering and evolutionary computing.



(a-1) f(PI, E_PI) (a-2) PI (a-3) E_PI

Fig. 8. The optimization process of each performance indexes for the linear fuzzy inference-based FNN model by the hybrid algorithm (No. of rules: 30, θ = 0.2).

Table 10. Comparison of performance with other modeling methods.

MODEL			PI	PI_s	E_PI_s	No. of rules
Ahn's Model [18]	FNN		5.835			7
	AIM		8.420			
Our Model	GA	θ=0.5		4.038	6.028	6
	Hybrid	θ=0.2		3.725	5.291	
		θ=0.5		3.830	5.397	

The HCM clustering is used here to determine initial values (in the strict sense, initial regions to be used in GA) of apexes of the membership functions of the information granules used in the model. The hybrid Identification algorithm is used for auto-tuning of the parameters of the FNN model such as apexes of the membership functions, learning rates, and momentum coefficients. The proposed hybrid identification algorithm is effective for nonlinear complex system, so we can construct a well-structured model. The introduced performance index helps achieve a balance between the approximation and generalization abilities. These two could be easily balanced by choosing a value for the weighting factor. The experimental studies clearly revealed that the models are compact (realized through a small number of rules) and we can obtain better performance (both at the level of approximation and generalization capabilities) for several commonly used experimental data sets.

REFERENCES

[1] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: A survey," *Fuzzy Sets Syst*., vol. 66, pp. 1-13, 1994.
[2] M. M. Gupta and D. H. Rao, "On the principles of fuzzy neural networks," *Fuzzy Sets Syst*., vol. 61, pp. 1-18, 1994.
[3] L. A. Zadeh, "Fuzzy sets," *Inf. Control 8*, pp. 338-353, 1965.
[4] L. A. Zadeh, "Outline of a new approach to the

analysis of complex systems and decision processes," *IEEE Trans. on Syst., Man, Cybern*., vol. 3, pp. 28-44, January 1973.

[5] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Norwell, MA: Kluwer, 1996.

[6] D. E. Golderg, *Genetic Algorithm in search, Optimization & Machine Learning,* Addison Wesley, 1989.

[7] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1992.

[8] K. A. De Jong, *Are Genetic Algorithms Function Optimizers?*, Parallel Problem Solving from Nature 2, Manner, R. and Manderick, B. eds., North-Holland, Amsterdam, 1992.

[9] S.-K. Oh and W. Pedrycz, "Identification of fuzzy systems by means of an auto-tuning algorithm and its application to nonlinear systems," *Fuzzy sets and Syst*., vol. 115, no. 2, pp. 205-230, 2000.

[10] D. E. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, California: Holden Day, 1976.

[11] G. Vachtsevanos, V. Ramani, and T. W. Hwang, *Prediction of Gas Turbine NOx Emissions using Polynomial Neural Network*, Technical Report, Georgia Institute of Technology, Atlanta, 1995.

[12] S.-K. Oh, K.-C. Yoon, and H.-K. Kim, "The design of optimal fuzzy-neural networks structure by means of ga and an aggregate weighted performance index," *Journal of Control, Automation and Systems Engineers*, vol. 6, no. 3, pp.273-283, 2000.

[13] S.-K. Oh, W. Pedrycz, and B.-J. Park, "Hybrid identification of fuzzy rule-based models," *Int. J. of Intelligent Systems,* vol. 17, no.1, pp. 77-103, Jan. 2002.

[14] E. Kim, H. Lee, M. Park, and M. Park, "A simply identified Sugeno-type fuzzy model via double clustering," *Information Sciences*, vol. 110, pp. 25-39. 1998.

[15] Y. Lin and G. A. Cunningham III, "A new approach to fuzzy-neural modeling," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 2, pp. 190-197, 1997.

[16] B.-J. Park, W. Pedrycz, and S.-K. Oh, "Identification of fuzzy models with the aid of evolutionary data granulation," *IEE Proc. CTA*, vol. 148, Issue 05, pp. 406-418, 2001.

[17] K.-C. Yoon and B. J. Park, and S.-K. Oh, "Fuzzy-neural networks based on improved fuzzy input space and its optimization," *Proc. of the 14th KACC*, vol. B, pp. 314-317, Oct, 1999(in Korean).

[18] T.-C. Ahn and S.-K. Oh, *Emission Pattern Model for the Atmosphere Pollution Material of a Power Plant*, Technical report, Electrical Engineering & Science Research Institute, Korea, 1997(in Korean).

[19] S.-K. Oh, *Fuzzy Model & Control System by* C-Programming, Naeha press, 2002.

[20] S.-K. Oh, *Computational Intelligence by Programming focused on Fuzzy, Neural Networks, and Genetic Algorithms*, Naeha Press, 2002.

Ho-Sung Park received the B.S. and M.S. degrees in Control and Instrumentation Engineering from the Wonkwang University, Korea in 1999 and 2001, respectively. He is currently a Ph.D. student at the same institute. His research interests include fuzzy and hybrid systems, neurofuzzy models, genetic algorithms, and computational intelligence. He is a member of ICASE and KIEE.

Sung-Kwun Oh received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983-1989, he worked as the Senior Researcher in the R&D Lab. of Lucky-Goldstar Industrial Systems Co., Ltd. He was a Postdoctoral Fellow in the Department of Electrical and Computer Engineering at the University of Manitoba, Canada, from 1996 to 1997. He is currently an Associate Professor in the School of Electrical, Electronic and Information Engineering, Wonkwang University, Korea. His research interests include fuzzy systems, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He is a member of IEEE. He currently serves as an Associate Editor of the Korean Institute of Electrical Engineers (KIEE) and the Institute of Control, Automation, and Systems Engineers (ICASE), Korea.