ELSEVIER

Advanced Engineering Informatics 16 (2002) 247–263

ADVANCED ENGINEERING INFORMATICS

www.elsevier.com/locate/aei

Implicit rule-based fuzzy-neural networks using the identification algorithm of GA hybrid scheme based on information granulation

Sung-Kwun Oh[a,*], Witold Pedrycz[b,c], Ho-Sung Park[a]

[a]*School of Electrical Electronic and Information Engineering, Wonkwang University, 344-2, Shinyong-Dong, Iksan, Chon-Buk 570-749, South Korea*
[b]*Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alta, Canada AB T6G 2G6*
[c]*Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland*

Received 10 September 2002; accepted 29 April 2003

Abstract

This paper proposes an identification method for nonlinear models realized in the form of implicit rule-based fuzzy-neural networks (FNN). The design of the model dwells on the technologies of computational intelligence (CI), namely fuzzy sets, neural networks, and genetic algorithm. The FNN modeling and identification environment realizes parameter estimation through a synergistic usage of clustering techniques, genetic optimization and a complex search method. An HCM (Hard C-Means) clustering algorithm helps determine an initial location (parameters) of the membership functions of the information granules to be used in this fuzzy model. The parameters such as apexes of membership functions, learning rates, and momentum coefficients are then adjusted using the optimization algorithm of a GA hybrid scheme. The proposed GA hybrid scheme combines GA with the improved complex method to guarantee both global optimization and local convergence. An aggregate objective function (performance index) is used in the model design in order to achieve a sound balance between its approximation and generalization abilities. The proposed type of the model is experimented with several time series data (gas furnace, sewage treatment process, and NO*x* emission process data of gas turbine power plant).

Keywords: Fuzzy-neural networks; Computational intelligence; HCM clustering; GA hybrid scheme; Genetic algorithm; Improved complex method

1. Introduction

During the past few years, fuzzy neural network (FNN) has 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 the two fields in which significant advances have been made in the last decade [1,2]. There have been many attempts to synthesize FNN to fully benefit from the advantages of these two technologies.

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 resulted in many fruitful results both in theory and application [5,16].

In the early approaches, the generation of the fuzzy rules and the adjustment of its membership functions were realized by *trial and error* and/or operator's experience. Subsequently, the designers find it difficult to develop adequate fuzzy rules and membership functions to reflect the essence of the data. Moreover, some information gets 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 sub optimal. 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 impressive panoply of neuro-fuzzy approaches, the comprehensive solution is still to be developed. Interestingly, in this synergistic arrangement, they tend to compensate disadvantages of these two technologies when used in the context of fuzzy rule-based models. The essential advantage of neural networks is in their adaptive nature and learning from historical data. In the context of rules, the learning concerns the parameters of the membership functions.

In this paper, we use implicit linear fuzzy inference-based FNN model as the fuzzy set-based approach by the fuzzy partitions formed for the individual variables. This

* Corresponding author.

1474-0346/02/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S1474-0346(03)00016-8

model comes with substantial learning speed and better convergence characteristics than other FNN models.

The FNN combines fuzzy 'if–then' rules with complementary membership functions with neural networks that are learned by means of the standards backpropagation algorithm [6]. The structure of the network is constructed by partitioning fuzzy input–output space for each input variable. This approach exhibits some limits that come with an inability to capture the essential characteristics of input–output data. To overcome such drawback and discuss all variables en bloc, we consider the use of a HCM (Hard C-Means) clustering algorithm. As this method concentrates on revealing the essential structure of the data, it helps form a blueprint (structural skeleton) of the network; more specifically we determine initial apexes of the membership functions used in the model. In the sequel, by using the identification algorithm of a GA hybrid scheme, we further optimize the model. The hybrid identification algorithm exploits the ideas of genetic algorithms (GAs) [9–11] and improved complex algorithm [12]. GAs are global optimization techniques that avoid many shortcomings existing in conventional search techniques when operating in large and complex problem spaces. Despite their successes reported in many publications, by combining these optimization tasks we end up with a problem that is highly nonlinear and may not fit well to 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. Genetic techniques have shown to be flexible meaning that they are capable of carrying out a comprehensive optimization of the parameters of the FNN model. However, they do not guarantee convergence to a global optimum. In order to solve this problem, we use improved complex algorithm that exploits the convergence of problem-specific technique.

We introduce an aggregate objective function [12] that takes into account both training data and validation 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 gas furnace process [13,14], activated sludge process in sewage treatment system [12] and NOx emission process data of gas turbine power plant [15].

2. Implicit 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 implicit type of 'if–then' rules along with their development mechanisms, that is, the linear fuzzy inference-based FNN as implicit rule base type. The inference scheme that is used in the conclusion part of the rules comes with a linear inference.

The conclusion of the rule is expressed in the form of a linear relationship between inputs and an output variable. The partition of the fuzzy input space being used in the model helps FNNs achieve faster learning produce better convergence characteristics.

The notation used in Fig. 1 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. Not necessarily f_i is a sigmoid function encountered conventional neural networks but we allow for more flexibility in this regard. Finally, the output of the FNN $\hat{y}$ is governed by the following expression

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

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 Eq. (1) as the following mappings (rules)

$$\begin{aligned} R^1 &: \text{IF } x_i \text{ is } A_{i1} \text{ then } Cy_{i1} = w0_{i1} + x_i wa_{i1} \\ &\vdots \\ R^j &: \text{IF } x_i \text{ is } A_{ij} \text{ then } Cy_{ij} = w0_{ij} + x_i wa_{ij} \\ &\vdots \\ R^n &: \text{IF } x_i \text{ is } A_{in} \text{ then } Cy_{in} = w0_{in} + x_i wa_{in} \end{aligned} \tag{2}$$

Being 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} as shown in Fig. 2. $w0_{ij}$ is a constant consequence of the rule and wa_{ij} is an input variable consequence of the rule. They express a connection (weight) existing between the neurons as we have already visualized in Fig. 1. Furthermore we confine ourselves to triangular membership functions and make their membership grades sum up to 1 (so this leads to a fuzzy partition of the variable). Each membership function in the premise part of fuzzy rule is assigned to be complementary with neighboring ones in the sense shown in Fig. 2.

Mapping from x_i to $f_i(x_i)$ is determined by the fuzzy inferences and a standard defuzzification. The inference result coming from Eq. (2) follows a standard center of gravity aggregation.

$$\begin{aligned} f_i(x_i) &= \sum_{j=1}^{n} \bar{\mu}_{ij}(x_i)(w0_{ij} + x_i wa_{ij}) \\ &= \frac{\sum_{j=1}^{n} \mu_{ij}(x_i)(w0_{ij} + x_i wa_{ij})}{\sum_{j=1}^{n} \mu_{ij}(x_i)} \end{aligned} \tag{3}$$

where $\bar{\mu}_{ij}(x_i)$ denotes the normalized version of $\mu_{ij}(x_i)$.

An input signal x_i activates only two membership functions labeled by k and $k+1$ simultaneously as shown

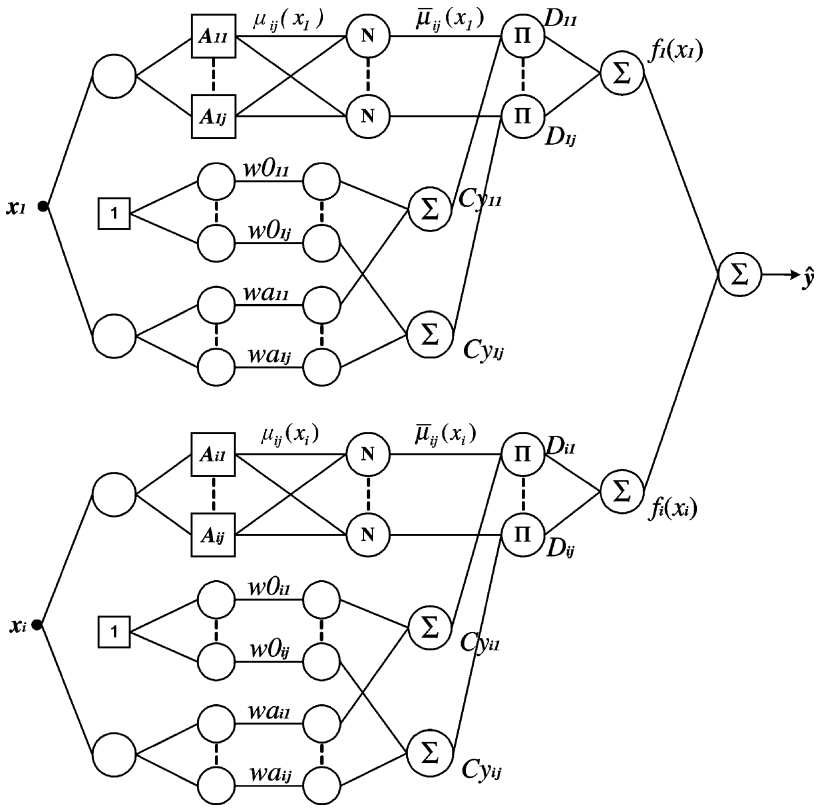


Fig. 1. Linear fuzzy inference-based FNNs structure.

in Fig. 2. The sum of the grades of these two neighboring membership functions, labeled by k and $k+1$, is always equal to 1, that is, $\mu_{ik}(x_i)+\mu_{ik+1}(x_i)=1$. In the sequel, Eq. (3) reduces to the form

$$f_i(x_i)=\mu_{ik}(x_i)(w0_{ik}+x_i wa_{ik})+\mu_{ik+1}(x_i)(w0_{ik+1}+x_i wa_{ik+1}) \tag{4}$$

The learning of FNN is realized by adjusting connections of the neurons and as such it follows a standard back-propagation (BP) algorithm, cf. Refs. [7,8]. In this study, we use the Euclidean error as a performance measure

$$E_p=(y_p-\hat{y}_p)^2 \tag{5}$$

where, E_p is an error observed for the p-th input–output data set $(\mathbf{x}_p,y_p)=(x_{1p},x_{2p},\ldots,x_{mp},y_p)$, $p=1,2,3,\ldots,n$ (with n being the total number of data points), y_p is the p-th target output data and $\hat{y}_p$ stand for the p-th actual output of the model for this specific data point. For n input–output data pairs, an overall (global) performance index is written down as a sum of the errors

$$E=\frac{1}{n}\sum_{p=1}^{n}(y_p-\hat{y}_p)^2 \tag{6}$$

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

$$wa(\text{new})=wa(\text{old})+\Delta wa \tag{7}$$

where the update formula follows the gradient descent method, namely

$$\Delta wa_{ij}=\eta_a\left(-\frac{\partial E_p}{\partial wa_{ij}}\right) \tag{8}$$

with η_a being a positive learning rate.

Moreover we have

$$-\frac{\partial E_p}{\partial wa_{ij}}=-\frac{\partial E_p}{\partial \hat{y}_p}\frac{\partial \hat{y}_p}{\partial f_i(x_i)}\frac{\partial f_i(x_i)}{\partial D_{ij}}\frac{\partial D_{ij}}{\partial Cy_{ij}}\frac{\partial Cy_{ij}}{\partial wa_{ij}} \tag{9}$$

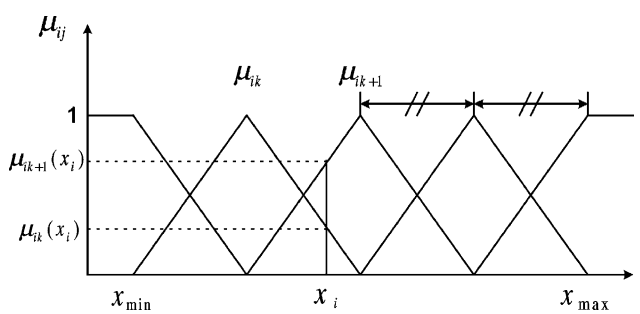


Fig. 2. Description of membership functions.

Each part of right side in Eq. (9) is expressed in the form

$$-\frac{\partial E_p}{\partial \hat{y}_p} = -\frac{\partial}{\partial \hat{y}_p}(y_p - \hat{y}_p)^2 = 2(y_p - \hat{y}_p),$$
$$\frac{\partial \hat{y}_p}{\partial f_i(x_i)} = 1, \quad \frac{\partial f_i(x_i)}{\partial D_{ij}} = 1, \quad \frac{\partial D_{ij}}{\partial Cy_{ij}} = \mu_{ij}(x_i), \tag{10}$$
$$\frac{\partial Cy_{ij}}{\partial wa_{ij}} = x_i$$

Finally, we obtain

$$\Delta wa_{ij} = 2\eta_a(y_p - \hat{y}_p)\mu_{ij}(x_i)x_i \tag{11}$$

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

$$m(t) = w(t) - w(t-1) \tag{12}$$

The complete update formula combining the already discussed components reads as

$$\Delta wa_{ij} = 2\eta_a(y_p - \hat{y}_p)\mu_{ij}(x_i)x_i + \alpha_a(wa_{ij}(t) - wa_{ij}(t-1)) \tag{13}$$

(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 problems 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 tuned.

Genetic algorithms are optimization techniques based on the principles of natural evolution. In essence, they are search algorithms that use operations found in natural genetic to guide the journey through a search space. GA's have been theoretically and empirically proven to provide robust search capabilities in complex spaces offering a valid approach to problems requiring efficient and effective searching.

Traditional GAs, though robust, is generally not the most successful optimization algorithm for any particular domain. That is, there is no guarantee that a 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. While being quite efficient, it depends very much on an initial values of the parameters. Therefore, if we start with some incorrect initial value, it may not converge to the local minimum. The sequence of usage of the HCM and GA gives rise to the formation of a promising search area (in which we can position some rough values of the parameters) that is followed by further refined search in the parameters space contributes to a well-defined and comprehensive optimization process.

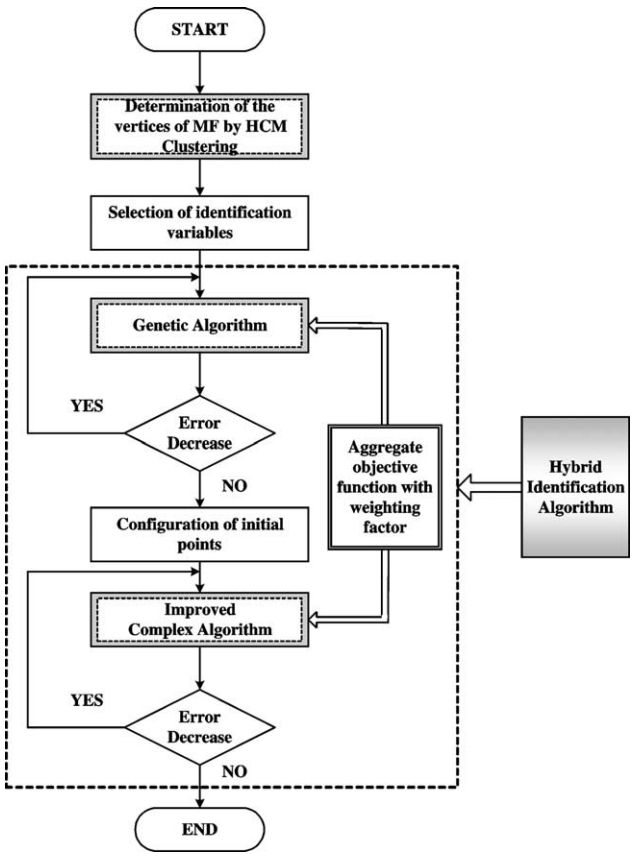


Fig. 3. A general flowchart of the hybrid identification algorithm outlining main development phases.

In this study, the proposed hybrid identification algorithm for dynamic parameters of fuzzy neural networks is sought, which combines the abilities of GAs and improved complex method thus resulting in an improved performance of the optimization process.

To determine suitable values of the parameters for a given model, a hybrid identification algorithm is developed. An overall flowchart of the design process indicating clearly how the optimization mechanisms of the FNN model are employed is visualized in Fig. 3.

3.1. HCM clustering method for defining the apexes of membership functions

When dealing with the premise parts of the rules, we confine ourselves to triangular type of membership function. For the triangular membership functions we have either two or three parameters to optimize. In the simplest scenario shown in Fig. 4, the maximal and minimal initial values of the vertical points of the membership functions depend on the range of experimental data encountered in the dataset.

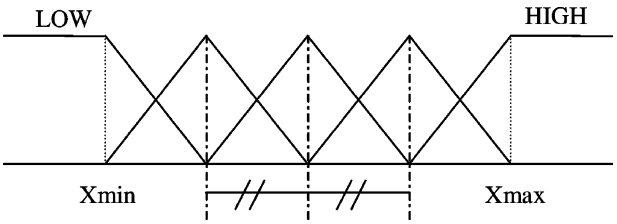


Fig. 4. Fuzzy partition composed of uniformly distributed fuzzy sets (information granules).

Note that in this case fuzzy sets are distributed uniformly across the entire universe of discourse (space). Owing to the properties of these fuzzy sets (uniform distribution), only the boundaries of the entire space are subject to optimization. The situation shown in Fig. 5 is quite different from the uniform distribution of the information granules. The clustering method (HCM) helps us organize the data into clusters and in this way we take into account the characteristics of the experimental data. In the regions where some clusters of data have occurred we end up with some fuzzy sets that help represent the specificity of the data set. In the sequel, the modal values of the clusters are refined (optimized) using genetic optimization (GA).

It is worth emphasizing that the HCM clustering method has been used extensively not only to organize and categorize data, but it becomes useful 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 $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_k = [x_{k1}, \ldots, x_{km}]$, 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 is the number of clusters. Each A_i is viewed as a cluster, so $\{A_1, \ldots, A_c\}$ 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

$$u_{ik} = \begin{cases} 1, & \mathbf{x}_k \in A_i \\ 0, & \mathbf{x}_k \notin A_i \end{cases} \tag{14}$$

where $\mathbf{x_k} \in \mathbf{X}$, $A_i \in P(X)$, $i = 1, 2, \ldots, n$. Clearly, $u_{ik} = 1$ means that $\mathbf{x}_k$ belongs to cluster A_i. Given the value of u_{ik},

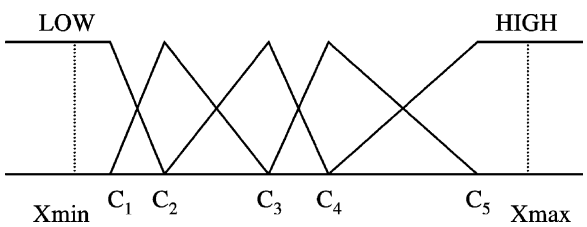


Fig. 5. Fuzzy partition composed of nonuniformly distributed fuzzy sets constructed with the aid of the clustering technique.

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

$$u_{ik} \in \{0, 1\}, \quad 1 \le i \le c, \quad 1 \le k \le n \tag{15}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \qquad \forall k \in \{1, 2, \ldots, n\} \tag{16}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \qquad \forall i \in \{1, 2, \ldots, c\} \tag{17}$$

At the interpretation end, Eqs. (15) and (16) mean that each $\mathbf{x}_k \in X$ should belong to one and only one cluster. Eq. (17) requires that each cluster A_i must contain at least one and at most $n - 1$ data point. Collecting u_{ik} with $1 \le i \le c$ and $1 \le k \le n$ into a $c \times n$ matrix $\mathbf{U} = [u_{ik}]$. We obtain the matrix representation for hard c-partition, defined as follows.

$$M_C = \left\{ \mathbf{U} | \boldsymbol{u}_{ik} \in \{0, 1\}, \quad \sum_{i=1}^{c} u_{ik} = 1, \quad 0 < \sum_{k=1}^{n} u_{ik} < n \right\} \tag{18}$$

[*Step* 1]: Fix 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:

$$\mathbf{v}_i^{(r)} = \{v_{i1}, v_{i2}, \ldots, v_{ij}, \ldots, v_{im}\} \tag{19}$$

$$v_{ij}^{(r)} = \frac{\sum_{k=1}^{n} u_{ik}^{(r)} x_{kj}}{\sum_{k=1}^{n} u_{ik}^{(r)}} \tag{20}$$

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

[*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

$$d_{ik} = d(\mathbf{x}_k - \mathbf{v}_i) = \|\mathbf{x}_k - \mathbf{v}_i\| = \left[\sum_{j=1}^{m} (x_{kj} - v_{ij})^2 \right]^{1/2} \tag{21}$$

$$\mathbf{u}_{ik}^{(r+1)} = \begin{cases} 1 & d_{ik}^{(r)} = \min\{d_{jk}^{(r)}\} \text{ for all } j \in c \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

[*Step* 4]: Check a termination criterion. If

$$\|\mathbf{U}^{(r+1)} - \mathbf{U}^{(r)}\| \le \varepsilon \text{ (tolerance level)} \tag{23}$$

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

3.2. Genetic algorithm

The need to handle optimization problems whose objective functions are complex and non-differentiable arises in many areas of system analysis and synthesis.

While there are a number of analytic and numerical optimization techniques aimed at solving these tasks, there exists a large class of problems that are out of reach by standard gradient-oriented techniques. Among objective functions which are highly challenging to these classical methods are those that are non-convex, multi-modal, and noisy [11].

Genetic algorithms [9–11] have proven to be useful in optimization of such problems because of their ability to efficiently use historical information to obtain new solutions with enhanced performance and a global nature of search supported there. Genetic algorithms are also theoretically and empirically proven to support robust search in complex search spaces. Moreover they do not get trapped in local minima as opposed to gradient decent techniques being quite susceptible to this shortcoming. GAs is population-based optimization techniques.

The search of the solution space is completed with the aid of several genetic operators. There are three basic operators used in any GA- supported search, namely selection, crossover, and mutation. Reproduction is a process in which the mating pool for the next generation is chosen. Individual strings are copied into the mating pool according to their fitness function values. Crossover usually proceeds in two steps. First, members from the mating pool are mated at random. Second, each pair of strings undergoes crossover as follows: a position l along the string is selected uniformly at random from the interval $[1, l-1]$, where l is the length of the string. Two new strings are created by swapping all characters between the positions k and l. Mutation is a random alteration of the value of a string position. In a binary coding, mutation means changing a zero to a one or vice versa. Mutation occurs with small probability. Those operators, combined with the proper definition of the fitness function, constitute the main body of the genetic computing.

In this paper, for the optimization of the FNN model, GAs use the serial method of binary type, roulette-wheel in the selection operator, one-point crossover in the crossover operator, and invert in the mutation operator. The GA used here employs the elitist strategy to assure that the best individual is retained when moving from population to population. In the experiment, we run the method for 100 generations, use 60 individuals, and implement a resolution of 10 bits per each variable, 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. Fig. 6 shows how a string is composed in GAs. Here, variable $x_1, x_2, \ldots$, and x_k denote input variables of the FNN model, η and α denote learning ratio and momentum coefficient, respectively, while c_{ij} ($i = 1, 2, \ldots$; variable no., $j = 1, 2, \ldots$; membership function no.) denotes the vertical point of membership functions for each input variable.

3.3. Improved complex algorithm

Usually, by combining these optimization tasks we end up with a problem that is highly nonlinear and may not fit well to 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 [12].

We realize 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

$$\text{Minimize } f(\mathbf{X}) \tag{24}$$

Subject to

$$g_j(\mathbf{X}) \le 0, \qquad j = 1, 2, \ldots, m \tag{25}$$

$$x_i^{(l)} \le x_i \le x_i^{(u)}, \qquad i = 1, 2, \ldots, n \tag{26}$$

where the superscripts l and u denote the lower and upper bound 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 the elements of the FNN model. They include the apexes of membership function, learning rates, and momentum coefficients.

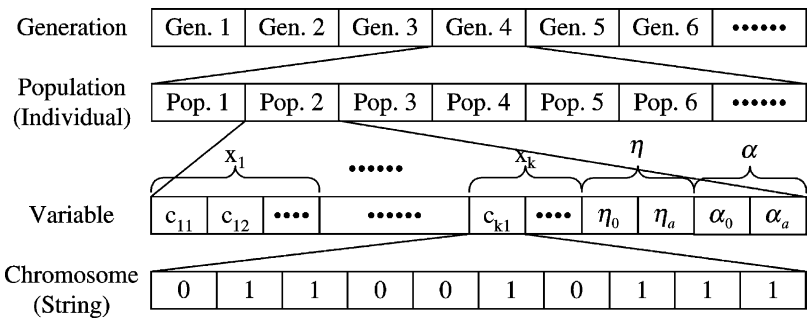


Fig. 6. Data structure of GA for optimization of FNN model.

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

⟨Step 2⟩

The initial values of α, γ and β is specified using the Reflection, Expansion and Contraction of simplex concept as follows

$$\text{(I) Reflection} : \mathbf{X}_r = \mathbf{X}_o + \alpha(\mathbf{X}_o - \mathbf{X}_h) \tag{27}$$

$$\text{(II) Expansion} : \mathbf{X}_e = \mathbf{X}_o + \gamma(\mathbf{X}_r - \mathbf{X}_o) \tag{28}$$

$$\text{(III) Contraction} : \mathbf{X}_c = \mathbf{X}_o + \beta(\mathbf{X}_h - \mathbf{X}_o) \tag{29}$$

⟨Step 3⟩

$\mathbf{X}_h$ and $\mathbf{X}_l$ are the vertices corresponding to the maximum function value $f(\mathbf{X}_h)$ and the minimum function value $f(\mathbf{X}_l)$. $\mathbf{X}_o$ is the centroid of all the points $\mathbf{X}_l$ except $i = h$. The reflection point $\mathbf{X}_r$ is given by Eq. (27), with $\mathbf{X}_h = \max f(\mathbf{X}_i)$, $(i = 1, 2, \ldots, k)$,

$$\mathbf{X}_o = \frac{1}{m-1}\left(M\sum_{i=1}^{m}\mathbf{X}_i - \mathbf{X}_h\right) \text{ and } \alpha = \frac{\|\mathbf{X}_r - \mathbf{X}_o\|}{\|\mathbf{X}_h - \mathbf{X}_o\|}$$

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

⟨Step 4⟩

If a reflection process gives a point $\mathbf{X}_r$ for which $f(\mathbf{X}_r) < f(\mathbf{X}_l)$, i.e. if the reflection produces a new minimum, we expand $\mathbf{X}_r$ to $\mathbf{X}_e$ by Eq. (28), with

$$\gamma = \frac{\|\mathbf{X}_e - \mathbf{X}_o\|}{\|\mathbf{X}_r - \mathbf{X}_o\|} > 1$$

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

⟨Step 5⟩

If the reflection process produces a point $\mathbf{X}_r$ for which $f(\mathbf{X}_r) > f(\mathbf{X}_i)$, for all i except $i = h$ and $f(\mathbf{X}_r) < f(\mathbf{X}_h)$, then we replace the point $\mathbf{X}_h$ by $\mathbf{X}_r$. In this case, we contract the simplex as in Eq. (29), with

$$\beta = \frac{\|\mathbf{X}_c - \mathbf{X}_o\|}{\|\mathbf{X}_h - \mathbf{X}_o\|}$$

If $f(\mathbf{X}_r) > f(\mathbf{X}_h)$, we use $\mathbf{X}_c$ without changing the previous point $\mathbf{X}_h$. If $\mathbf{X}_c$ does not satisfy the constraints, a new point $\mathbf{X}_c$ is generated with $\mathbf{X}_c = (\mathbf{X}_o + \mathbf{X}_c)/2$. This process is conducted repeatedly until $\mathbf{X}_c$ satisfies the constraints. If the contraction process produces a point $\mathbf{X}_c$ for which $f(\mathbf{X}_c) < \min[f(\mathbf{X}_h), f(\mathbf{X}_r)]$, we replace the point $\mathbf{X}_h$ by $\mathbf{X}_c$. And proceed with the reflection again. On the other hand, if $f(\mathbf{X}_c) \geq \min[f(\mathbf{X}_h), f(\mathbf{X}_r)]$, we replace all $\mathbf{X}_i$ by $(\mathbf{X}_i + \mathbf{X}_l)/2$, and start 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:

$$Q = \left\{\sum_{i=1}^{m}\frac{[f(\mathbf{X}_i) - f(\mathbf{X}_o)]^2}{m}\right\}^{1/2} \leq \varepsilon \tag{30}$$

If Q does not satisfy Eq. (30), 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.

The flowchart of the proposed improved complex method is included in Fig. 7.

3.4. Identification algorithm by GA hybrid scheme

GAs are global optimization techniques that avoid many shortcomings exhibited in conventional search techniques when completed in a large and complex space. However, GAs carry out 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 [9]. While 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 problem with this method is about a selection of a starting point.

To alleviate these difficulties, we consider the hybrid identification algorithm. It combines genetic algorithm effectively with the improved complex method to guarantee both global optimization and local convergence. The features of the hybrid identification algorithm are described as follows.

1. GA can determine optimal parameters in a vast search space. The improved complex method can find the optimal parameters of the FNN within a limited region or a boundary condition, that is to say, when calculating activation degrees of each rule by the improved complex method through a vast searching space, overflow is appeared (happened) very often by generating '0', because activation degrees of linguistic labels by input of process dataset exceed a boundary region of membership parameters adjusted by the improved complex method.
2. GAs are an efficient tool for finding a global minimum area, but there is no guarantee that GAs will give the best solution in this area(region); usually we end up with the value that will be a near-optimal solution. The improved complex method is an efficient tool for finding an optimal solution considering a limited search region.

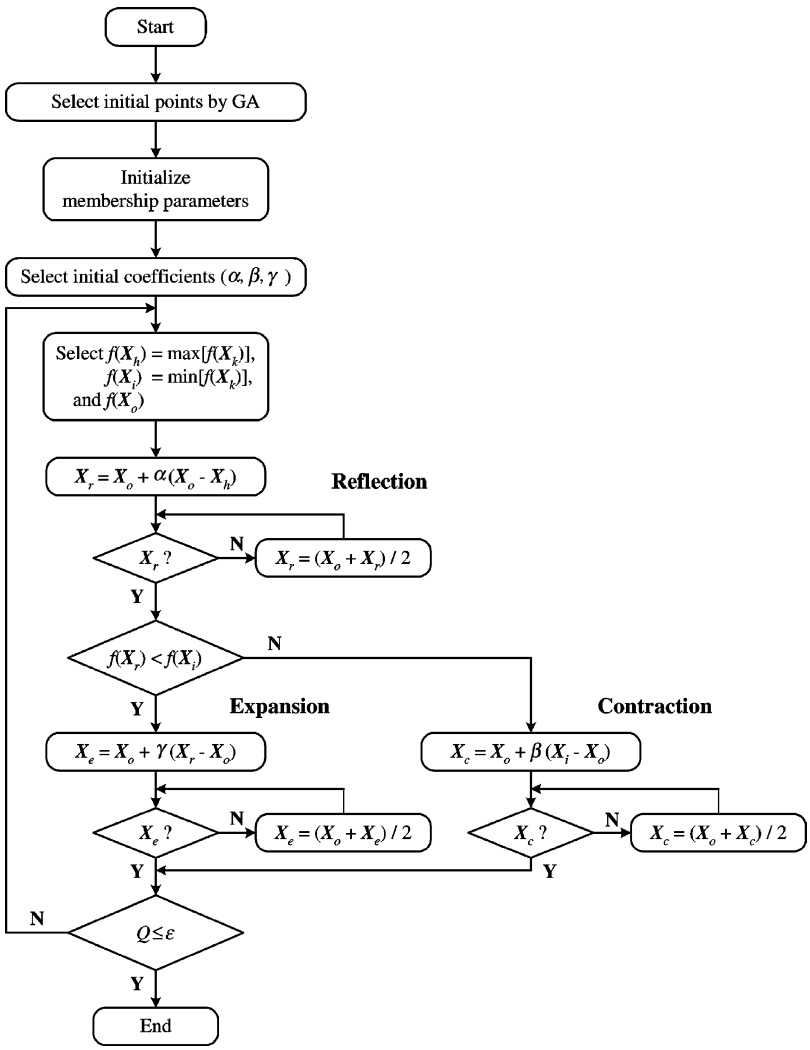


Fig. 7. Flowchart of improved complex algorithm for the parameters identification of FNN model.

3. GAs, which are optimization techniques based on the principles of biological evolution, approach effectively to optimal parameters in a vast searching space. But the improved complex method based on geometrical concept has difficulty in finding optimal parameters in case that initial values are over a limited region or a boundary condition. Therefore, following the hybrid structure combined with the two optimization methods of GAs and improved complex method, we can compute the auto-tuned parameters (membership parameters of the linguistic labels, learning ratio, and momentum coefficient.

Hybrid identification algorithm takes the advantage of GAs and 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

Conventional methods of system modeling construct the models on a basis of some training data and then evaluate it through the use of the validation data. In other words, the training data is used only for the model construction of the target process and the validation data is employed to evaluate the model performance. There is no guarantee that the required performance is met because the developed model is customized only for the training data. We call this aspect as an over-fitting. Consequently, the overfitting phenomenon can generate significant approximation errors and reduce further use of the model as a sound predictor. Therefore, the following objective function (or cost function) is employed to decrease the error and to increase the predictability (generalization) capability of the model - that is, the objective function includes the performance

index for training (*PI*), the performance index for evaluation (*E_PI*) that are combined by means of some weighting factor θ.

The objective function (performance index) is a basic instrument guiding the evolutionary search in the solution space [12]. The objective function includes both the training data and validation data and comes as a convex sum of these two components.

$$f(\mathrm{PI}, \mathrm{E\_PI}) = \theta \times \mathrm{PI} + (1 - \theta) \times \mathrm{E\_PI}(\mathrm{V\_PI}) \tag{31}$$

PI and E_PI (or V_PI) denote the performance index for the training and validation data, respectively. Moreover θ is a weighting factor that allows us to strike a balance between the performance of the model for the training and validation data. Depending upon the values of the weighting factor, several specific cases of the objective function are worth distinguishing.

Case 1. $\theta = 1 : f(\mathrm{PI}, \mathrm{E\_PI}) = \mathrm{PI}$

In this case, the objective function becomes $f(\mathrm{PI}, \mathrm{E\_PI}) = \mathrm{PI}$. Model optimization is based on the training data and the validation data is not considered. This case shows outstanding approximation capability but prediction (generalization) capabilities may become quite low relative to the approximation aspects of the model.

Case 2. $\theta = 0 : f(\mathrm{PI}, \mathrm{E\_PI}) = \mathrm{E\_PI}$

In this case, the objective function becomes $f(\mathrm{PI}, \mathrm{E\_PI}) = \mathrm{E\_PI}$. The model is constructed by the training data and then optimized from the viewpoint of E_PI that is obtained by the validation data. As a result, this method leads to slightly lower approximation abilities that are offset by higher generalization aspects of the model (relative to the first case discussed above).

Case 3. $\theta = 0.5 : f(\mathrm{PI}, \mathrm{E\_PI}) = 0.5\mathrm{PI} + 0.5\mathrm{E\_PI}$

Here both PI and E_PI are considered of equal relevance. In comparison to Case 1, this objective function implies lower approximation and generalization abilities; both these criteria are treated as equal (that may be a certain solution but not necessarily the best one) from the optimization standpoint.

Case 4. $\theta = \alpha(\alpha \in [0, 1]) : f(\mathrm{PI}, \mathrm{E\_PI}) = \theta \times \mathrm{PI} + (1 - \theta) \times \mathrm{E\_PI}$

Both PI and E_PI considered and the proper selection of θ establish the direction of optimization where we maintain balance between the approximation and generalization abilities. In this case, PI is obtained by the training data and E_PI is obtained from the validation data of the model constructed by the training data. Model selection is performed from the minimization of this aggregate objective function through the adjustment (optimization) of parameters related to FNNs.

When we consider the complex system with strong nonlinearities, note that if the weighting factor θ is selected smaller value than in Case 3, then the objective function will be nearly equivalent to Case 2 (that is, $f(\mathrm{PI}, \mathrm{E\_PI}) \cong \mathrm{E\_PI}$), and in this case the effect of PI (approximation ability) will be relatively decreased (that is, $\mathrm{PI} \ll \mathrm{E\_PI}$). On the other hand, if the weighting factor θ is selected as quite bigger value than in case of Case 3, note that the effect of the objective function will be nearly equivalent to Case 1 ($f(\mathrm{PI}, \mathrm{E\_PI}) \cong \mathrm{PI}$) and in this case the effect of E_PI (generalization ability) will be relatively decreased ($\mathrm{PI} \gg \mathrm{E\_PI}$). Therefore this method considers both approximation and generalization capability, and selects the proper model that satisfies required performance by selecting the adequate weighting factor θ.

The performance index used in the ensuing numerical experiments will be as Euclidean distance; see Eq. (6) while θ is regarded as an extra parameter.

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 [14], sewage treatment process [12], and NO*x* emission process data of gas turbine power plant [15].

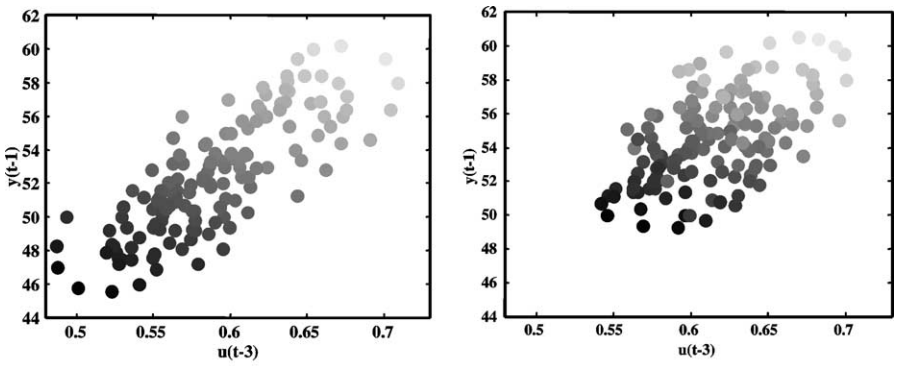


Fig. 8. Data points induced by I/O data set $[u(t-3), y(t-1); y(t)]$.

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

GA		Improved complex algorithm	
Generation	100	α	1
Population	60	β	0.5
String	10	γ	2
Crossover rate	0.6	ε	1×10^{-6}
Mutation probability	0.1	Complex iterations	500
FNN iterations	300	FNN iterations	100

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 [14]. We try to model the gas furnace using 296 pairs of input–output data. The total data set consisting 296 input–output pairs was split into two parts. The first one (consisting of 148 pairs) was used for training. The remaining part of the series serves as a validation set. 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:

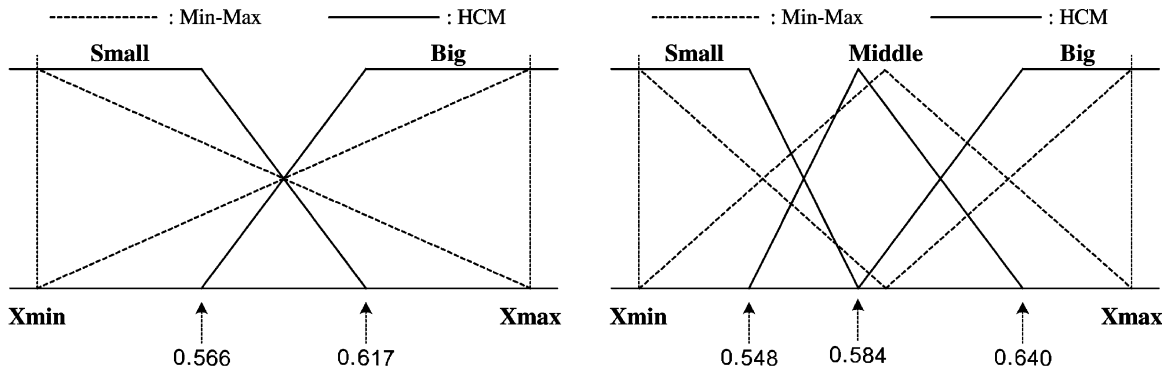
$$u(t) = 0.6 - 0.048 \times u_m(t) \tag{32}$$

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 $(u(t-3), y(t-1))$ and one-output $(y(t))$. Also, the total data set of gas furnace process is partitioned two parts, the training data and validation data.

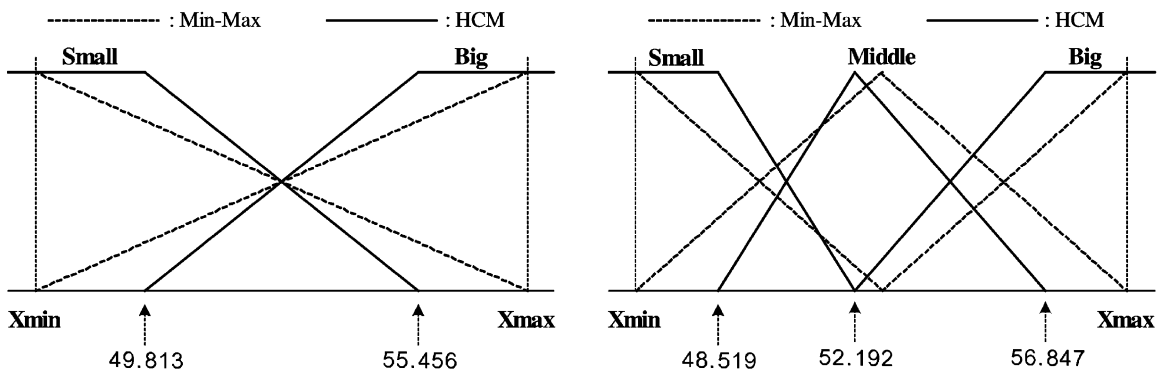
From the two-dimensional plot of the data set shown in Fig. 8, in the case of the training data and validation data, the data set $[u(t-3), y(t-1); y(t)]$ [12,20] exhibit more uniform and less sparse distribution than any other data set such as $[u(t-1), y(t-1); y(t)], \ldots, [u(t-5), y(t-1); y(t)], [y(t-3), y(t-2); y(t)]$, and $[u(t-3), y(t-3); y(t)]$. Table 1 summarizes computational aspects of the approach as well as elaborates on the remaining parameters of the hybrid algorithm.

Fig. 9 shows the membership functions of each input variable according to the partition of fuzzy input spaces by



(a-1) No. of membership functions : 2 (a-2) No. of membership functions : 3

(a) u(t-3)



(b-1) No. of membership functions : 2 (b-2) No. of membership functions : 3

(b) y(t-1)

Fig. 9. Definition of initial membership functions of gas furnace process by HCM.

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

Input variables		Weight factor (θ)	No. of membership functions					
gx1	gx2		2:2		3:2		3:3	
			PI	E_PI	PI	E_PI	PI	E_PI
$u(t-3)$	$y(t-1)$	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

a Min–Max method and the HCM clustering method. Just to mention that the Min-Max method uses the minimum and maximum value of experimental data encountered in the dataset.

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 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. 10. Moreover, in conjunction to Fig. 6, the optimized parameters of the membership functions by GA and hybrid algorithm are shown in Fig. 11.

Fig. 12 shows 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 and validation set. The relations of input–output data in gas furnace process as previously shown in Fig. 8 are feeble. Therefore the proposed hybrid algorithm gives rise to the similar results as encountered when applying GAs. Here PI denotes a performance index of the model for the training data set while E_PI describes the performance of the model for the validation data.

4.2. Sewage treatment process

Sewage treatment generally uses the activated sludge process that consisted of sand basin, primary sedimentation basin, aeration tank and final sedimentation basin (see Fig. 13). The suspended solid (SS) included in sewage is sedimented by gravity in sand and primary sedimentation basins. Air is consecutively absorbed in sewage in the aeration tank for several hours. Microbe lump (that is called floc or activated sludge) springing naturally, mainly remove

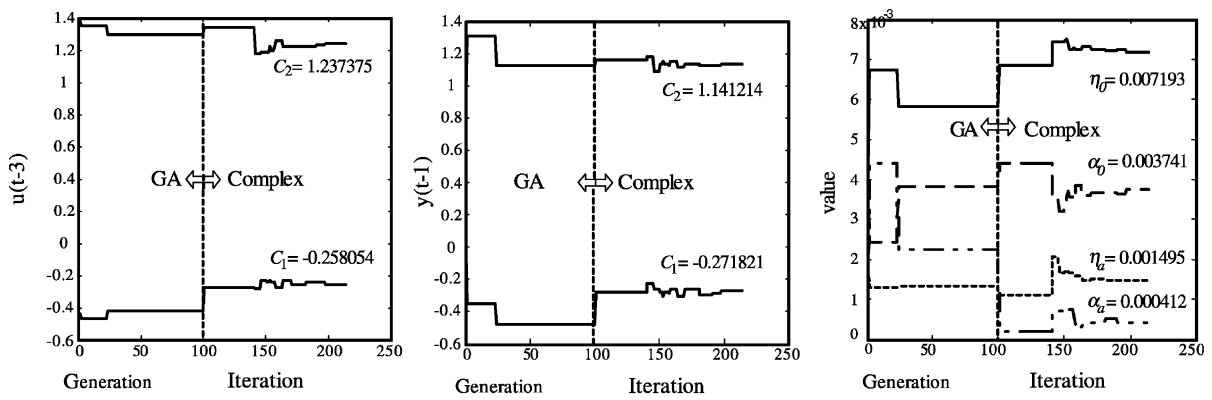


Fig. 10. Convergence process of optimal parameters for the linear fuzzy inference-based FNN model by Hybrid Algorithm ($\theta = 0.8$).

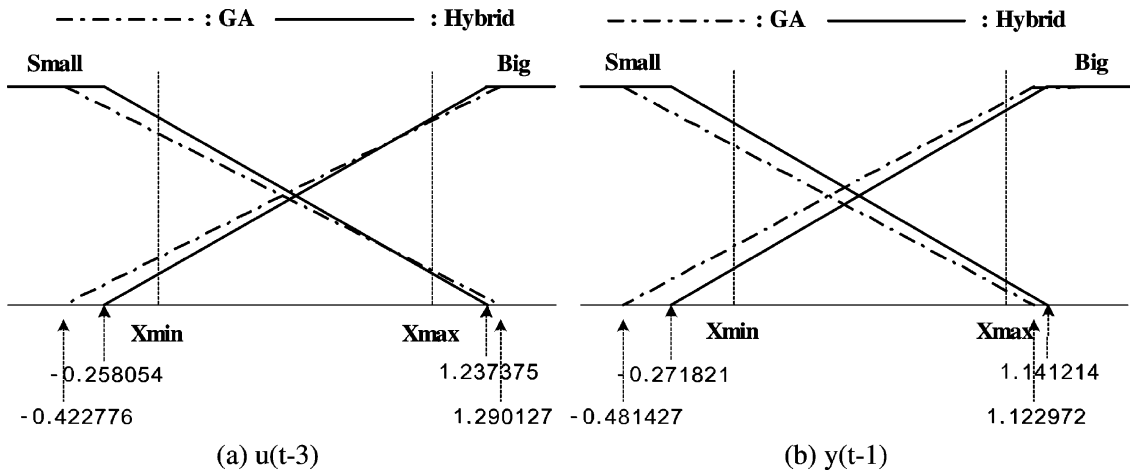


Fig. 11. The final tuned values of membership functions by GA and hybrid algorithm ($\theta = 0.8$).

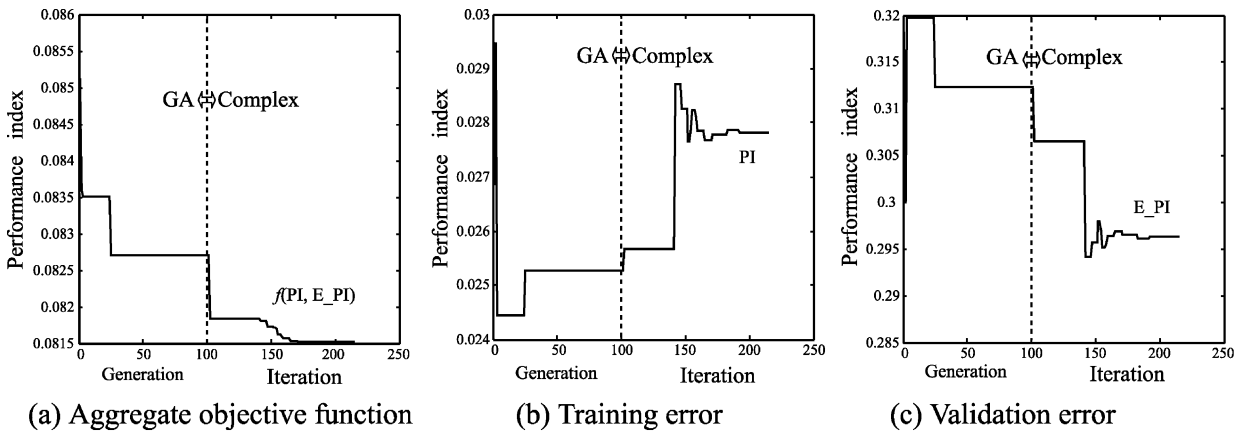


Fig. 12. The optimization process of each performance index for the linear fuzzy inference-based FNN model by the Hybrid Algorithm (No. of rules: 4, $\theta = 0.8$).

the organic matters in aeration tank. Activated sludge biochemical oxygenates, proliferates and resolve the organic matters into hydrogen and carbon dioxide by metabolism. In the final sedimentation basin, floc is sedimented, recycled and again used to remove the organic matters and then purified water is transported to tertiary sedimentation basin.

The activated sludge process is the process that involves an aeration tank and final sedimentation. We measure the biological oxygen demand (BOD) and the concentration of SS in influent sewage at primary sedimentation basin, and effluent BOD (EBOD) and SS (ESS) in effluent sewage at final sedimentation basin. Because EBOD and ESS are changed, dependent on BOD and SS, dissolved oxygen

Table 3
Comparative summary of existing fuzzy models

Model			PI	E_PI	No. of rules
Oh and Pedrycz's model [12]	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 [17]			0.034	0.244	2
Lin and Cunningham's model [18]			0.071	0.261	4
Our model	GA	$\theta = 0.5$	0.042	0.302	6
	GA + complex (hybrid)	$\theta = 0.8$	0.027	0.296	4
			0.028	0.294	5
		$\theta = 0.5$	0.029	0.296	6

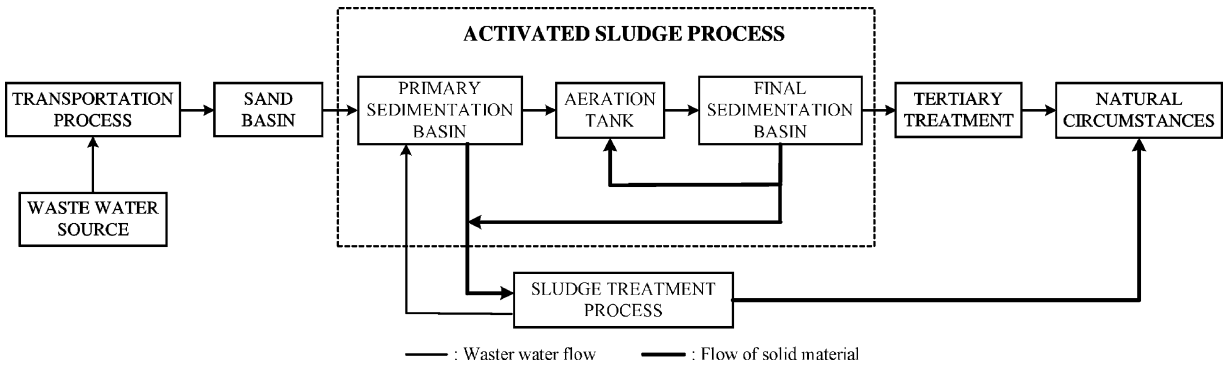


Fig. 13. Configuration of the sewage treatment system.

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

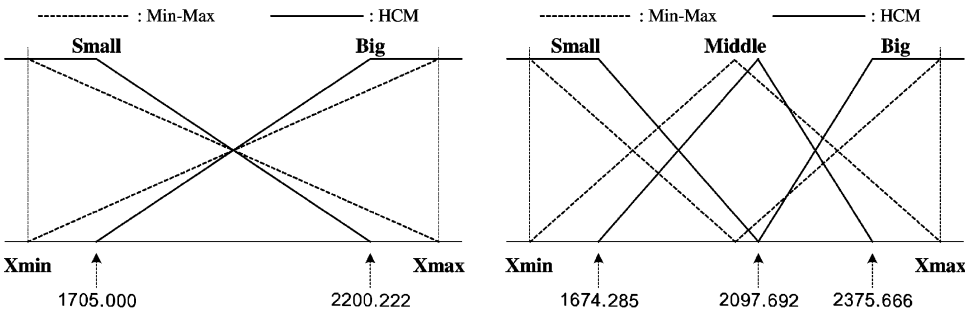
GA		Improved complex algorithm	
Generation	100	α	1
Population	60	β	0.5
String	10	γ	2
Crossover rate	0.6	ε	1×10^{-6}
Mutation probability	0.1	Complex iterations	500
FNN iterations	500	FNN iterations	500

set-point (DOSP) and recycle sludge ratio set-point (RRSP) are set so that ESS and EBOD should be kept up less than the prescribed small quantity. EBOD and ESS depend on mixed liquid suspended solid (MLSS), waste sludge ratio (WSR), RRSP and DOSP. BOD has a correlation with SS.

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 process [12].

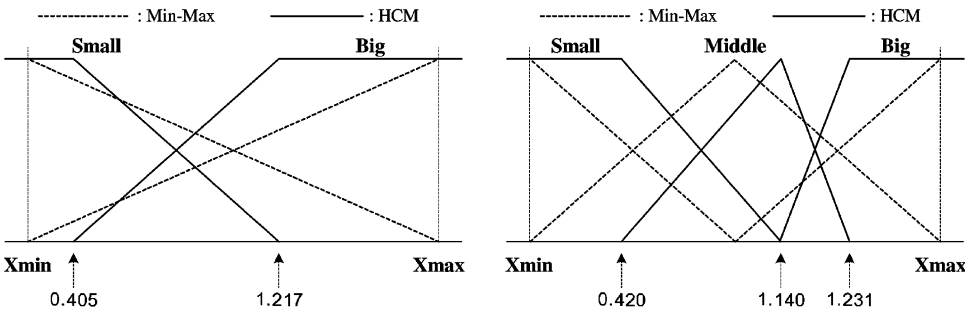
From four input variables (MLSS, WSR, RRSP, and DOSP), we choose two input variables that minimize the evaluation, and extract more than two fuzzy partitions (fuzzy sets LOW and HIGH) from each input–output pair of data. Table 4 shows computational cost and the related parameters used in the hybrid algorithm.

Fig. 14 shows membership functions of two inputs variable (MLSS and WSR) according to the partition of fuzzy input spaces by a Min–Max method and the HCM clustering method.



(a-1) No. of membership functions : 2 (a-2) No. of membership functions : 3

(a) MLSS



(b-1) No. of membership functions : 2 (b-2) No. of membership functions : 3

(b) WSR

Fig. 14. Definition of initial membership functions of sewage treatment process by HCM.

Table 5
Performance index of sewage treatment process according to the change of number of membership functions

Input variables		Weight factor (θ)	No. of membership functions (total no. of rules)					
gx1	gx2		2:2 (4)		3:2 (5)		3:3 (6)	
			PI	E_PI	PI	E_PI	PI	E_PI
MLSS	WSR	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 5 shows the values of the performance index of the FNN obtained using 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, see Fig. 15.

In case when the weighting factor and the number of rules are selected to 0.5 and 6, respectively, the vertical points (C_1, C_2, and C_3) of membership functions of each input variable, learning rate (η), and momentum coefficient (α) are converged to constant values through the hybrid

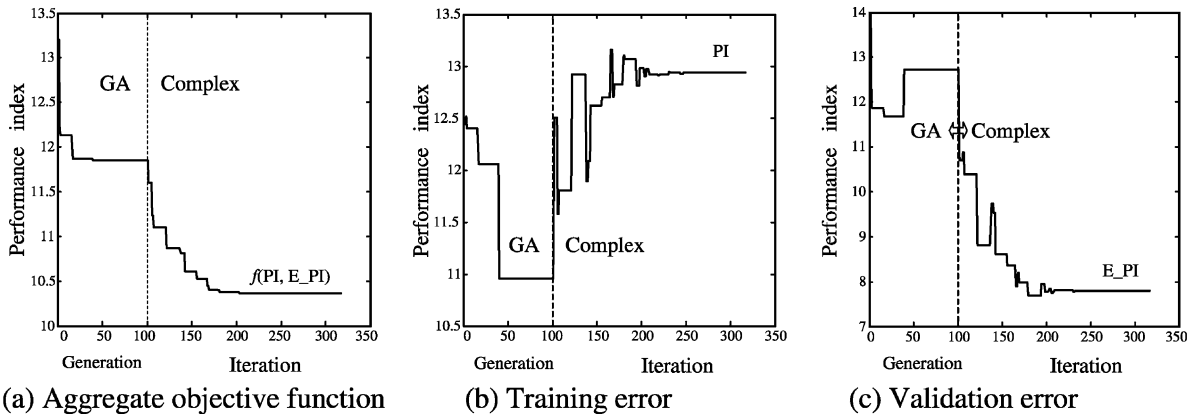


Fig. 15. The optimization process of each performance indexes for the linear fuzzy inference-based FNN model by the hybrid algorithm (No. of rules: 6, $\theta = 0.5$).

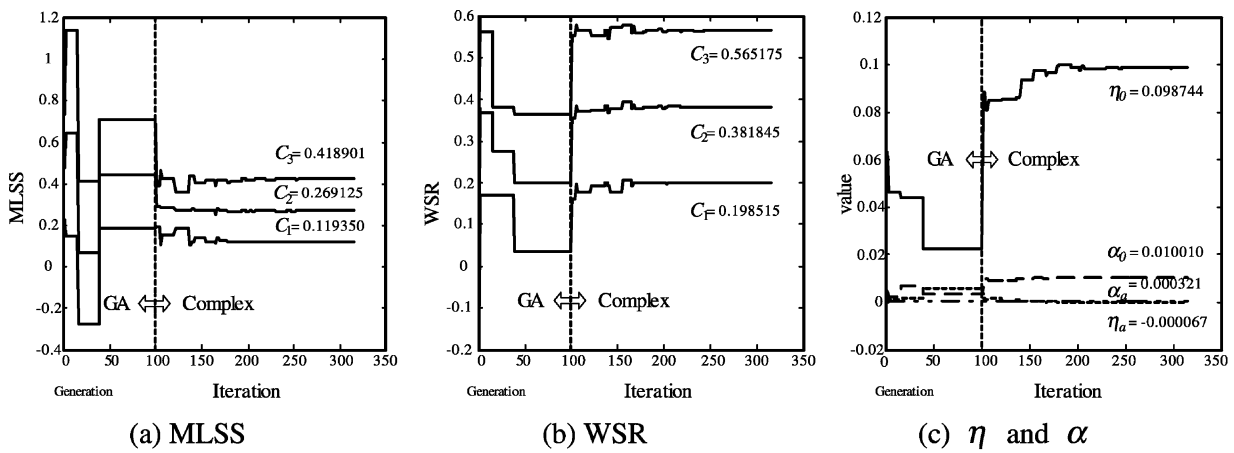


Fig. 16. Convergence process of optimal parameters for the linear fuzzy inference-based FNNs model by Hybrid Algorithm ($\theta = 0.5$).

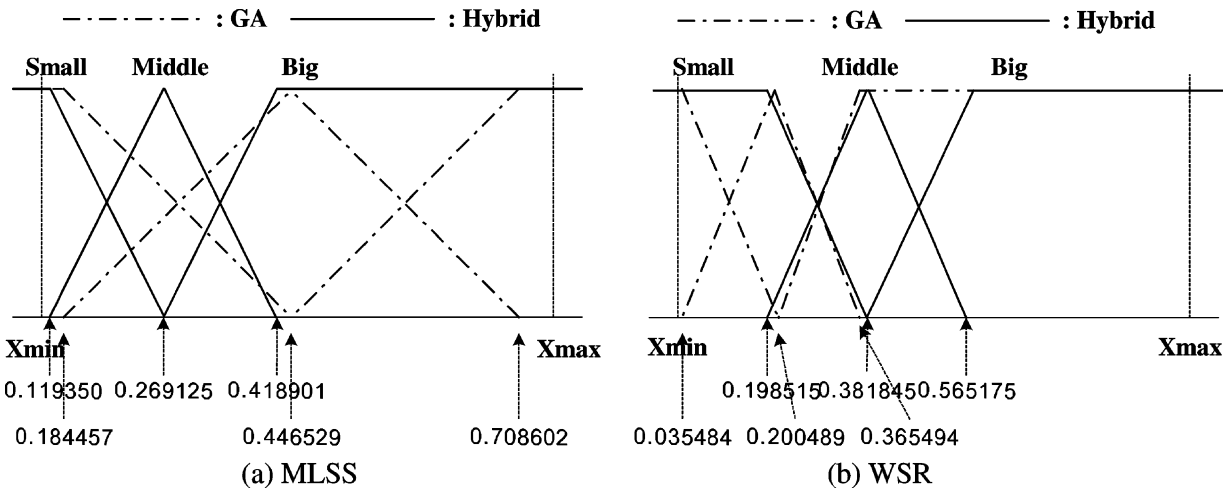


Fig. 17. The final tuned values of membership functions by GA and hybrid algorithm ($\theta = 0.5$).

optimization process as shown in Fig. 16. The optimized parameters of membership functions by GA and hybrid algorithm are included in Fig. 17.

Table 6 summarizes the results of comparative analysis of 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. Especially, Table 6 shows that in comparison with the PI, the E_PI (performance index for validation data) of the FNN model obtained using the hybrid identification algorithm is far better than in case when using GA. As we know from Table 6, we can obtain better performance through 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

NOx emission process is also modeled using the data of gas turbine power plants. Till now, almost NOx emission processes are based on 'standard' mathematical model in order to obtain regulation data from control process. However, such models do not develop the relationships between variables of the NOx emission process and parameters of its model in an effective manner. A NOx emission process of a GE gas turbine power plant located in Virginia, USA, is chosen in this modeling study.

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 [12]	Simplified		13.72	16.20	4
			14.10	16.56	6
			12.80	15.91	8
	Linear		6.39	54.23	4
			1.46	8.06×10^4	6
			0.001	923.32	8
Fuzzy model [19]	Simplified		12.35	11.17	6
	Linear		0.001	126.91	6
FNN model [20]			13.75	9.24	4
			12.34	10.19	9
Our model	GA	$\theta = 0.5$	12.59	10.22	4
			10.96	12.72	6
	GA + complex (hybrid)	$\theta = 0.5$	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

GA		Improved complex algorithm	
Generation	100	α	1
Population	60	β	0.5
String	10	γ	2
Crossover rate	0.6	ε	1×10^{-6}
Mutation probability	0.1	Complex iterations	500
FNN iterations	1000	FNN iterations	500

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

No. of MFs per input	Performance index, 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		3.640	6.468

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 [15]. The performance index is defined by Eq. (6).

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

$$y = -163.77341 - 0.06709x_1 + 0.00322x_2 + 0.00235x_3 + 0.26365x_4 + 0.20893x_5 \quad (33)$$

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 case of NOx emission process data, they have many input variables and a quantity of lots data. To gain a better insight into the performance characters of this process, an overall dataset (260 pairs of I/O data) is split into three parts, namely the training dataset (100 pairs of I/O data), validation dataset (100 pairs of I/O data) and testing dataset (60 pairs of I/O data). By using these dataset, we get the values of performance index according to the change of 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.

Here, PIs, V_PIs and E_PIs denote the performance index for the training dataset, validation dataset and validation 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 six MFs per input variable, the variation ratio (slope) of the performance index of the FNN model does not change radically—especially E_PIs, is 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 validation 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 better than the one we obtain through genetic optimization. The hybrid identification algorithm improves the performance of the linear fuzzy inference-based FNN model in comparison with GA.

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. 18 shows the optimization process by showing the values of the performance index in successive cycles of the hybrid algorithm.

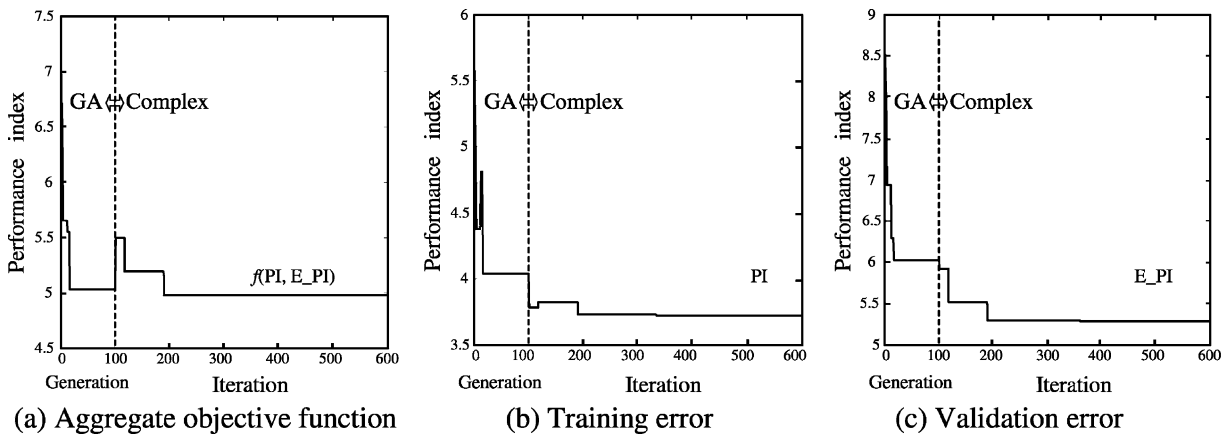


Fig. 18. The optimization process of each performance indexes for the linear fuzzy inference-based FNN model by the hybrid algorithm (No. of rules: 30, $\theta = 0.2$).

5. Concluding remarks

In this paper, the efficient identification technique is presented which 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 implicit 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.

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 FNN model such as apexes of the membership functions, learning rates, and momentum coefficients. The hybrid identification algorithm combines GAs with the improved complex method to guarantee both global optimization and local convergence. The experimental studies revealed that we can obtain better performance through the hybrid identification algorithm in NOx emission process data of heavy nonlinearity than uniformly distributed gas furnace data. In a word, 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 of 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.

Acknowledgements

This paper was supported by Wonkwang University in 2003.

References

[1] Buckley JJ, Hayashi Y. Fuzzy neural networks: a survey. Fuzzy Sets Syst 1994;66:1–13.
[2] Gupta MM, Rao DH. On the principles of fuzzy neural networks. Fuzzy Sets Syst 1994;61:1–18.
[3] Zadeh LA. Fuzzy sets. Inf Control 1965;8:338–53.
[4] Zadeh LA. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans Syst, Man, Cybern 1973; 3:28–44.
[5] Zimmermann H-J. Fuzzy set theory and its applications. Norwell, MA: Kluwer; 1996.
[6] Yamakawa T. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. Proceedings of the Second International Conference on Fuzzy logic and Neural Networks; 1992. p. 477–83.
[7] Oh S-K, Yoon K-C, Kim H-K. The design of optimal fuzzy-neural networks structure by means of GA and an aggregate weighted performance index. J Control, Automat Syst Engrs 2000;6(3):273–83.
[8] Oh S-K, Kim D-W, Park B-J. Advanced polynomial neural networks architecture with new adaptive nodes. Trans Control, Automat Syst Engng 2001;3(1):43–50.
[9] Golderg DE. Genetic algorithm in search, optimization and machine learning. Reading, MA: Addison-Wesley; 1989.
[10] Michalewicz Z. Genetic algorithms + data structures = evolution programs. Berlin: Springer; 1992.
[11] Jong D, KA. Are genetic algorithms function optimizers?, parallel problem solving from nature 2, Manner R, Manderick B, (Eds)., Amsterdam: North-Holland
[12] Oh S-K, Pedrycz W. Identification of fuzzy systems by means of an auto-tuning algorithm and its application to nonlinear systems. Fuzzy Sets Syst 2000;115(2):204–30.
[13] Kim D-W, Oh S-K, Kim H-K. A study on the self-organizing fuzzy polynomial neural networks. KIEE Int Trans Syst Control 2001;11(2): 79–89.
[14] Box DE, Jenkins GM. Time series analysis, forcasting and control. California: Holden Day; 1976.
[15] Vachtsevanos G, Ramani V, Hwang TW. Prediction of gas turbine NOx emissions using polynomial neural network. Technical Report, Georgia Institute of Technology, Atlanta; 1995
[16] Pedrycz W. An identification algorithm in fuzzy relational system. Fuzzy Sets Syst 1984;13:153–67.
[17] Kim E, Lee H, Park M, Park M. A simply identified Sugeno-type fuzzy model via double clustering. Inf Sci 1998;110:25–39.
[18] Lin Y, Cunningham III GA. A new approach to fuzzy-neural modeling. IEEE Trans Fuzzy Syst 1997;3(2):190–7.
[19] Park BJ, Pedrycz W, Oh SK. Identification of fuzzy models with the aid of evolutionary data granulation. IEE Proc—CTA 2001;148(5): 406–18.
[20] Oh S-K, Pedrycz W, Park B-J. Hybrid identification of fuzzy rule-based models. Int J Intell Syst 2002;17(1):77–103.