Fuzzy Relation-Based Fuzzy Neural-Networks Using a Hybrid Identification Algorithm

Ho-Sung Park and Sung-Kwun Oh

Abstract: In this paper, we introduce an identification method in Fuzzy Relation-based Fuzzy Neural Networks (FRFNN) through a hybrid identification algorithm. The proposed FRFNN modeling implement system structure and parameter identification in the efficient form of "If…, then…" statements, and exploit the theory of system optimization and fuzzy rules. The FRFNN modeling and identification environment realizes parameter identification through a synergistic usage of genetic optimization and complex search method. The hybrid identification algorithm is carried out by combining both genetic optimization and the improved complex method in order to guarantee both global optimization and local convergence. An aggregate objective function with a weighting factor is introduced to achieve a sound balance between approximation and generalization of the model. The proposed model is experimented with using two nonlinear data. The obtained experimental results reveal that the proposed networks exhibit high accuracy and generalization capabilities in comparison to other models.

Keywords: Fuzzy relation-based fuzzy neural networks, simplified and linear fuzzy inference, hybrid identification, genetic algorithms, improved complex method, aggregate objective function.

1. INTRODUCTION

As is widely known, both fuzzy logic systems and neural network systems are aimed at exploiting human-like knowledge processing capability. Recently, fuzzy logic systems and neural networks have been shown to obtain successful results in system information can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analyses. Much research has been done on applications of fuzzy neural network (FNN) systems, which combine the capability of fuzzy reasoning in handing uncertain information and the capability of neural networks in learning from processes [1-3]

In the early approaches, the generation of the fuzzy rules and the adjustment of its membership functions were done 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 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 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 relation-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 consider an extension of the network by considering the fuzzy partition realized in terms of fuzzy relations. That is, the structure of the network is constructed by partitioning fuzzy input-output space using all variables simultaneously. The networks are classified into the two main categories according to the type of fuzzy inference. We distinguish between a simplified and linear fuzzy inference. The FRFNN combines fuzzy "if-then" rules with neural networks that are learned by means of the standards back-propagation (BP). And using the hybrid identification algorithm, we further optimize the FRFNN model. The hybrid identification algorithm

Manuscript received December 9, 2002; accepted May 9, 2003. This work was supported by the Korea Science and Engineering Foundation (KOSEF) under Grant R05-2000-000-00284-0.

Ho-Sung Park is with the Department of Electrical Electronic and Information Engineering, Wonkwang University, Korea (e-mail: neuron@wonkwang.ac.kr).

Sung-Kwun Oh is with the Department of Electrical Electronic and Information Engineering, Wonkwang University, Korea (e-mail: ohsk@wonkwang.ac.kr).

dwells on the ideas of genetic algorithms (GAs) [4-6] and improved complex algorithm [7,8]. GAs is 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 autotuning 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 FRFNN model. However, they do not guarantee convergence to a global optimum. So to speak more specifically they help determine just initial regions (intervals) of the membership functions used in the model. 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 [7] 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 NOx emission process data of gas turbine power plant [9] and activated sludge process in sewage treatment system [7].

2. FUZZY RELATION-BASED FUZZY-NEURAL NETWORKS

The structure of FRFNN emerges at a junction of fuzzy sets and neural networks. In this section, we discuss two types of "if-then" rules along with their development mechanisms. We use fuzzy spaces partitioning in terms of all variables based on fuzzy relation-based approach. We distinguish between two classes (categories) of basic models. One uses a so-called simplified inference scheme that is used in the conclusion part of the rules. In the second case the conclusion part comes with a linear inference.

2.1. Simplified fuzzy inference-based FRFNN

Let us consider an extension of the network by considering the fuzzy partition realized in terms of fuzzy relations. The fuzzy partitions formed for the all variables lead us to the topology visualized in Fig. 1. Fig. 1 illustrates architecture of such FRFNN in case of two inputs and single output, where each input assumes three membership functions. The "circles" denote units of the FRFNN, the neuron denoted by $\prod$ realizes a Cartesian product. The outputs of these neurons are taken as a product of all the incoming signals. The "N" identifies a normalization procedure applied to the outputs taken as a product of membership grades. The "Σ" neuron is described by linear sum.

Making use of the language of the rule-based systems, the structure translates into the following collection of rules.

$$R^i : If\ x_1\ is\ A_{i1}\ and\ \cdots\ x_k\ is\ A_{ik},\ then\ y_i = w_i. \tag{1}$$

The fuzzy rules in (1) constitute an overall network of the FRFNN as shown in Fig. 1. The output f_i of each node generates a final output $\hat{y}$ of the form

$$\hat{y} = \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} \overline{\mu}_i \cdot w_i = \sum_{i=1}^{n} \frac{\mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i}. \tag{2}$$

The learning algorithm in FRFNN is realized by adjusting connection weights w_i of the neurons and as such it follows a standard Back-Propagation (BP) algorithm. We use the Euclidean error as a performance measure.

$$E_p = (y_p - \hat{y}_p)^2, \tag{3}$$

where E_p is an error 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 (global) performance index comes as a sum of the errors.

$$E = \frac{1}{N} \sum_{p=1}^{N} (y_p - \hat{y}_p)^2. \tag{4}$$

As far as learning is concerned, the connections change as follows:

$$w(new) = w(old) + \Delta w, \tag{5}$$

where the update formula follows the gradient descent method.

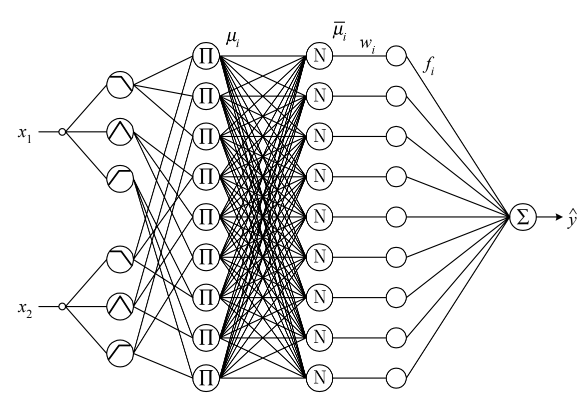


Fig. 1. Simplified fuzzy inference-based FRFNN structure.

$$\Delta w_i = \eta \cdot \left(-\frac{\partial E_P}{w_i} \right) = -\eta \cdot \frac{\partial E_p}{\partial \hat{y}_p} \cdot \frac{\partial \hat{y}_p}{\partial f_i} \cdot \frac{\partial f_i}{\partial w_i}. \tag{6}$$

Each part of right side in (6) is expressed in the form,

$$-\frac{\partial E_p}{\partial \hat{y}_p} = -\frac{\partial}{\partial \hat{y}_p}(y_p - \hat{y}_p)^2 = 2(y_p - \hat{y}_p),$$
$$\frac{\partial \hat{y}_p}{\partial f_i} = 1, \quad \frac{\partial f_i}{\partial w_i} = \bar{\mu}_i. \tag{7}$$

Therefore, Δw_i summarizes as follows:

$$\Delta w_i = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i \tag{8}$$

with η being a positive learning rate.

Quite commonly to accelerate convergence, a momentum term is being added to the learning expression. Combining (9) and a momentum term, we have,

$$\Delta w_i = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i + \alpha(w_i(t) - w_i(t-1)). \tag{9}$$

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

2.2. Linear fuzzy inference-based FRFNN

The conclusion is expressed in the form of a linear relationship between inputs and output variable. In case of linear inference-based FRFNN, the model of the proposed FNN comes in the form shown in Fig. 2.

Making use of the language of the rule-based systems, the structure translates into the following collection of rules.

$$R^i: \textit{If } x_1 \textit{ is } A_{i1} \textit{ and } \cdots x_k \textit{ is } A_{ik}, \quad \textit{then } Cy_i = wa_{0i} + x_1 wa_{1i} + \cdots + x_k wa_{ki}. \tag{10}$$

The fuzzy rules in (10) constitute an overall network

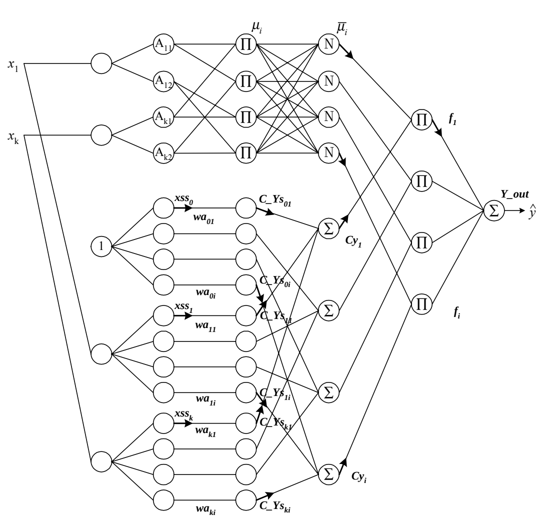


Fig. 2. Linear fuzzy inference-based FRFNN structure.

of the FRFNN as shown in Fig. 2. The output f_i of each node generates a final output $\hat{y}$ of the form

$$\hat{y} = \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} \bar{\mu}_i \cdot Cy_i = \sum_{i=1}^{n} \frac{\mu_i \cdot (wa_{0i} + x_1 wa_{1i} + \cdots + x_k wa_{ki})}{\sum_{i=1}^{n} \mu_i}. \tag{11}$$

The learning algorithm in FRFNN is realized by adjusting connection weights wa_{0i} and wa_{ki} of the neurons and as such it follows a standard BP algorithm. We use the Euclidean error as a performance measure.

$$E_p = (y_p - \hat{y}_p)^2, \tag{12}$$

where E_p is an error 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 (global) performance index comes as a sum of the errors.

$$E = \frac{1}{N} \sum_{p=1}^{N} (y_p - \hat{y}_p)^2. \tag{13}$$

As far as learning is concerned, the connections change as follows:

$$wa(new) = wa(old) + \Delta wa, \tag{14}$$

where the update formula follows the gradient descent method.

$$\Delta wa_{0i} = \eta \cdot \left(-\frac{\partial E_P}{wa_{0i}} \right) = -\eta \cdot \frac{\partial E_p}{\partial \hat{y}_p} \cdot \frac{\partial \hat{y}_p}{\partial f_i} \cdot \frac{\partial f_i}{\partial Cy_i} \cdot \frac{\partial Cy_i}{\partial C\_Ys_{0i}} \cdot \frac{\partial C\_Ys_{0i}}{\partial wa_{0i}}, \tag{15}$$

$$\Delta wa_{ki} = \eta \cdot \left(-\frac{\partial E_P}{wa_{ki}} \right) = -\eta \cdot \frac{\partial E_p}{\partial \hat{y}_p} \cdot \frac{\partial \hat{y}_p}{\partial f_i} \cdot \frac{\partial f_i}{\partial Cy_i} \cdot \frac{\partial Cy_i}{\partial C\_Ys_{ki}} \cdot \frac{\partial C\_Ys_{ki}}{\partial wa_{ki}}. \tag{16}$$

Each part of right side in (15) and (16) are expressed in the form,

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

$$\frac{\partial \hat{y}_p}{\partial f_i} = 1, \quad \frac{\partial f_i}{\partial Cy_i} = \mu_i, \quad \frac{\partial Cy_i}{\partial C\_Ys_{0i}} = 1,$$

$$\frac{\partial C\_Ys_{0i}}{\partial wa_{0i}} = xss_0 = 1, \tag{17}$$

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

$$\frac{\partial f_i}{\partial Cy_i} = \mu_i, \quad \frac{\partial Cy_i}{\partial C\_Ys_{ki}} = 1, \quad \frac{\partial C\_Ys_{ki}}{\partial wa_{ki}} = xss_k. \tag{18}$$

Therefore, Δwa_{0i} and Δwa_{ki} summarize as follow:

$$\Delta wa_{0i} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i, \tag{19}$$

$$\Delta wa_{ki} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i \cdot xss_k, \tag{20}$$

with η being a positive learning rate.

Quite commonly to accelerate convergence, a momentum term is being added to the learning expression. Combining (19), (20) and a momentum term, we have,

$$\Delta wa_{0i} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i + \alpha(wa_{0i}(t) - wa_{0i}(t-1)), \tag{21}$$

$$\Delta wa_{ki} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i \cdot xss_k + \alpha(wa_{ki}(t) - wa_{ki}(t-1)). \tag{22}$$

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

3. OPTIMIZATION OF FRFNN BY THE HYBRID IDENTIFICATION ALGORITHM

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. GAs 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. But it has difficult problem about selection of initial value.

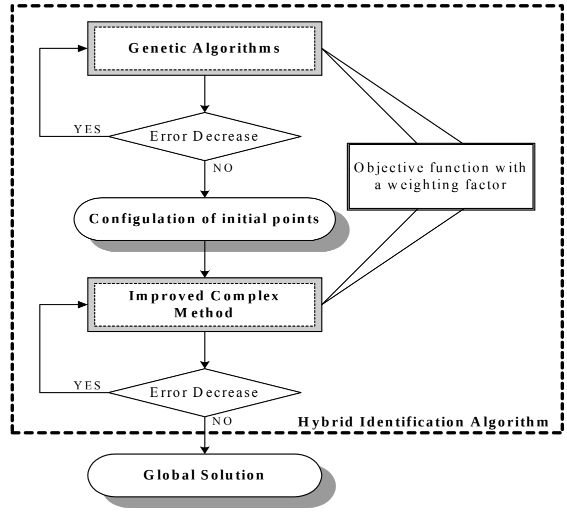


Fig. 3. A general flowchart of the hybrid identification algorithm outlining main development phases.

Therefore, if we select incorrect initial value, it may not converge to the local minimum point.

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

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

3.1. Genetic algorithms

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 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 [5].

Genetic algorithms [4-6] 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 genetic operators used in any GA- supported search, that is reproduction, 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 FRFNN 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. Here, we use 100 generations, 60 populations, 10 bits per string, 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.

3.2. Improved complex algorithms

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 [7].

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.

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

subject to

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

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

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.

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

$$\text{I) Reflection}: \boldsymbol{X}_r = \boldsymbol{X}_o + \alpha(\boldsymbol{X}_o - \boldsymbol{X}_h), \tag{26}$$

$$\text{II) Expansion}: \boldsymbol{X}_e = \boldsymbol{X}_o + \gamma(\boldsymbol{X}_r - \boldsymbol{X}_o), \tag{27}$$

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

<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, …,$k$), $\mathbf{X}_o = \frac{1}{m-1}(\sum_{i=1}^{n}\mathbf{X}_i - \mathbf{X}_h)$ and $\alpha = \frac{\|\mathbf{X}_r - \mathbf{X}_o\|}{\|\mathbf{X}_h - \mathbf{X}_o\|}$.

If $\boldsymbol{X}_r$ may 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 $\boldsymbol{X}_r$ satisfies 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 $\boldsymbol{X}_e$ satisfies 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 start 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 $\boldsymbol{X}_c$ satisfies 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 $\boldsymbol{X}_c$. And proceed with the reflection again. On the other hand, if $f(\boldsymbol{X}_c) \geq \min[f(\boldsymbol{X}_h), f(\boldsymbol{X}_r)]$, we replace all $\boldsymbol{X}_i$ by $(\boldsymbol{X}_i + \boldsymbol{X}_l)/2$, 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:

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

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 $\alpha = 1$, $\beta = 0.5$, and $\gamma = 2$, respectively.

3.3. Identification algorithms by hybrid scheme

GAs is global optimization techniques that avoid many shortcomings exhibited in conventional search techniques when completed in a large and complex space. However, GAs are a blind search and does 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 [4]. 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.

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.

3.4. 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 testing data. In other words, the training data is used only for the model construction of the target process and the testing 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 [7]. The objective function includes both the training data and testing data (or validation data) and comes as a convex sum of two components.

$$f(PI, E\_PI) = \theta \times PI + (1-\theta) \times E\_PI(V\_PI). \quad (30)$$

PI and *E_PI* (or *V_PI*) denote the performance index for the training data and testing data (or 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 testing data. Depending upon the values of the weighting factor, several specific cases of the objective function are worth distinguishing.

♦ **Case 1**. θ=1: *f*(*PI, E_PI*)=*PI*

In this case, the objective function becomes *f*(*PI, E_PI*)=*PI*. Model optimization is based on the training data and the testing data is not considered. This case shows outstanding approximation capability but predictable capability (or generalization) become lower relatively to approximation capability.

♦ **Case 2**. θ=0 : *f*(*PI, E_PI*)=*E_PI*

In this case, the objective function becomes *f*(*PI, E_PI*)=*E_PI*. The model is constructed by the training data and then optimized from the viewpoint of E_PI that is obtained by the testing data. The effect of this method shows low approximation capability but predictable capability (or generalization) is relatively increased in comparison to Case 1.

♦ **Case 3**. θ=0.5 : *f*(*PI, E_PI*)=0.5*PI* + 0.5*E_PI*

Both PI and E_PI are considered in equal. The effect of this objective function shows relatively lower approximation capability than in case of Case 1 and also it shows lower predictable capability (Generalization) than in case of Case 2.

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

Both PI and E_PI considered and the proper selection of θ establish the direction of optimization to maintain balance between the approximation and generalization. In this case, PI is obtained by the training data and E_PI is obtained from the testing 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 FRFNN.

4. EXPERIMENTAL STUDIES

Once the identification methodology has been established, one can proceed with intensive experimental studies. In this section, we provide two numerical examples to evaluate the advantages and the effectiveness of the proposal approach. These include NOx emission process data of a gas turbine power plant [9] and sewage treatment process [7].

4.1. 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, U. S. A., is chosen in this modeling 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 [9]. The performance index is defined by (3). We consider 260 pairs of the original input-output data. 130 out of 260 pairs of input-output data are used as training data set; the remaining part serves as a testing data set.

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

$$y=-163.77341-0.06709x_1+0.00322x_2 \\ +0.00235x_3+0.26365x_4+0.20893x_5\,. \tag{31}$$

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 1 shows computational cost and the related parameters used in the hybrid identification algorithm.

In case of NOx emission process data, they have many input variables and a quantity of lots data. To look into the performance characters of this process, overall dataset pairs of I/O data are 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 two.

Table 2 includes the values of the performance index of the FRFNN model derived when using the hybrid identification. 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, the performance index for the linear inference method-based FRFNN is better than the one produced by the simplified inference method-based FRFNN. Also, 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.

Table 1. Parameters of the optimization environment and computational effort.

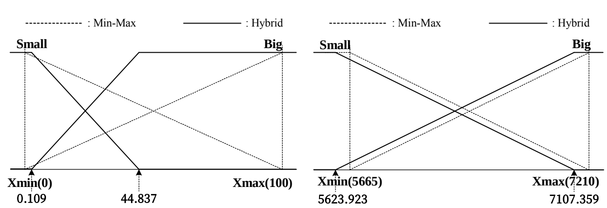
Genetic Algorithms		Improved Complex Algorithm	
Generation	100	α	1
Population	60	β	0.5
String	10	γ	2
Crossover rate	0.6	ε	1×10^{-6}
Mutation probability	0.1	Complex iterations	500
FNN iterations	1000	FNN iterations	1000

Fig. 4 shows the membership functions of each input variable according to the partition of fuzzy input spaces by a Min-Max method and the hybrid identification algorithm. Just to mention that the Min-Max method uses the minimum and maximum value of experimental data encountered in the dataset.

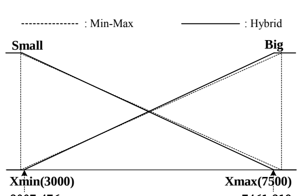
Fig. 5 illustrates the optimization process by visualizing the performance index in successive cycles (generation and iterations) of the hybrid identification algorithm. It also shows the preferred network architectures.

Table 2. Performance index as a function of the weighting factor.

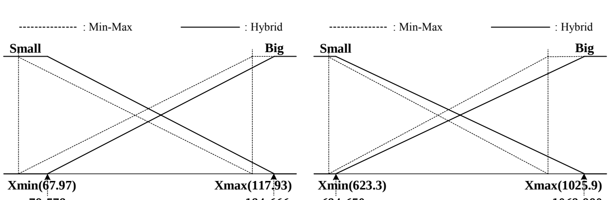
θ	Simplified inference method		Linear inference method	
	PI	E_PI	PI	E_PI
0.0	0.7103	1.6443	0.0802	0.1901
0.25	0.7051	1.6463	0.0802	0.1901
0.5	0.7001	1.6498	0.0802	0.1902
0.75	0.6912	1.6758	0.0629	0.2386
1.0	0.6804	1.7479	0.0617	0.2476



(a-1) AT (a-2) CS

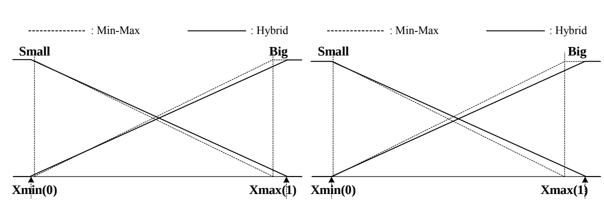


(a-3) LPTS

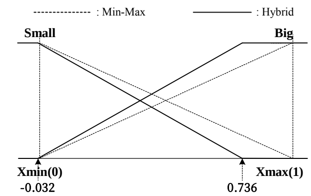


(a-4) CDP (a-5) TET

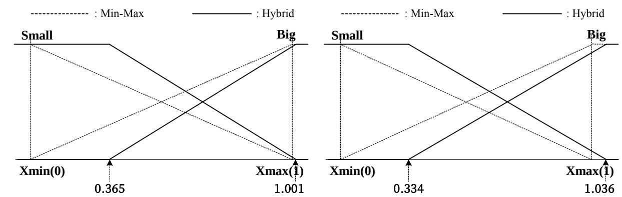
(a) Simplified inference method-based FRFNN (θ=0.5).



(b-1) AT (b-2) CS



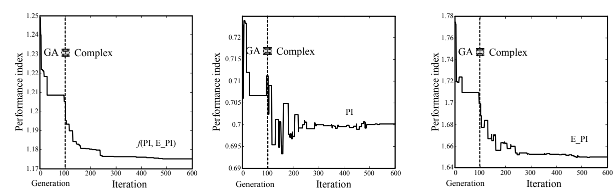
(b-3) LPTS



(b-4) CDP (b-5) TET

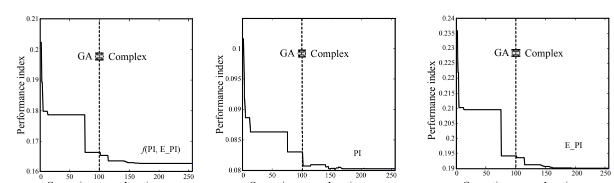
(b) Linear inference method-based FRFNN (θ=0.25).

Fig. 4. The final tuned values of membership functions by hybrid identification algorithm.



(a-1) $f(PI, E\_PI)$ (a-2) PI (a-3) E_PI

(a) Simplified inference method-based FRFNN (θ=0.5).



(b-1) $f(PI, E\_PI)$ (b-2) PI (b-3) E_PI

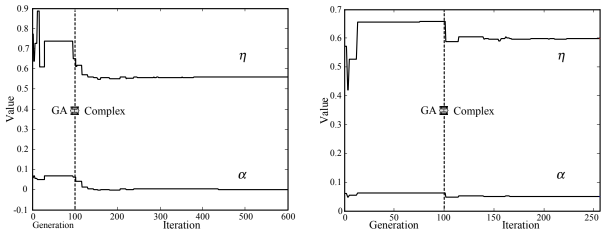
(b) Linear inference method-based FRFNN (θ=0.25).

Fig. 5. The optimization process of each performance indexes by the hybrid identification algorithm.

Fig. 6 shows the optimization process by visualizing learning rate and momentum coefficient in successive cycles (generation and iterations) of the hybrid identification algorithm. The original output and model output are shown in Fig. 7. The errors of relation-based fuzzy-neural networks are shown in Fig. 8.

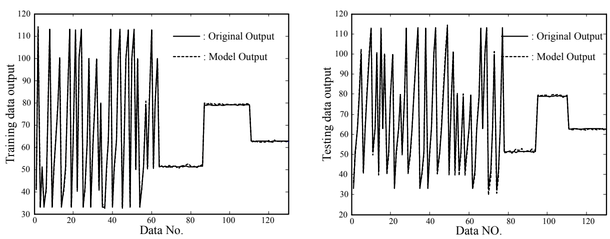
4.2. Sewage treatment process

Sewage treatment generally uses the activated sludge process that consisted of sand basin, primary sedimentation basin, aeration tank and final sedimentation basin (see Fig. 9). The suspended solid included in sewage is sedimented by gravity in sand

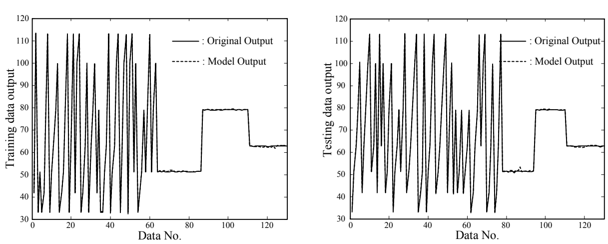


(a) Simplified (b) Linear

Fig. 6. The search process to optimal parameters by hybrid identification algorithm (θ = 0.25).



(a) Simplified inference method-based FRFNN (θ=0.5).



(b) Linear inference method-based FRFNN (θ=0.25).

Fig. 7. Original output and model output of relation-based fuzzy-neural networks.

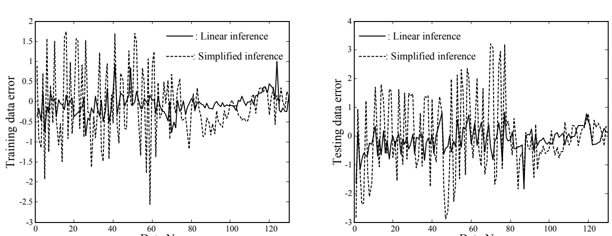


Fig. 8. Errors curves of relation-based fuzzy-neural networks.

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 the organic matters in aeration tank. Activated sludge biochemically 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

Table 3. Comparison of performance with other modeling methods.

MODEL			PI	PI_s	E_PI_s
Regression model				17.68	19.23
Ahn's Model [10]	Neural Networks		1773.3		
	FNN		5.835		
	AIM		8.420		
Fuzzy set-based FNN [11]	Simplified	θ=0.4		6.269	8.778
	Linear	θ=0.2		3.725	5.291
Multi-FNN [12]	Simplified	θ=0.5		2.806	5.164
	Linear	θ=0.75		0.720	2.025
Our model	Simplified	θ=0.5		0.700	1.649
	Linear	θ=0.25		0.080	0.190

the concentration of Suspended Solid (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 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 [7]. From four input variables (MLSS, WSR, RRSP, and DOSP), we choose two input variables (MLSS and 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 cost and the related parameters used in the hybrid identification algorithm.

As illustrated in Table 5, the performance index for the linear inference method-based FRFNN is better than the one produced by the simplified inference method-based FRFNN.

The Fig. 10 shows membership functions of two inputs variable (MLSS and WSR) according to the partition of fuzzy input spaces using a Min-Max method and the hybrid algorithm method.

The proposed model has 4 rules membership functions as shown in Fig. 10 (a), (b).

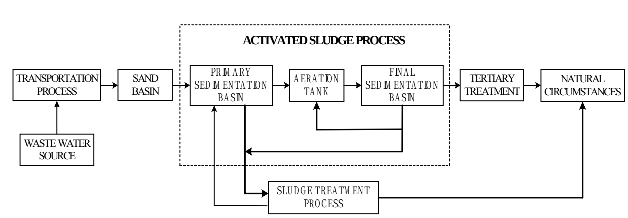


Fig. 9. Configuration of the sewage treatment system.

Fig. 11 illustrates the optimization process by visualizing the performance index in successive cycles (generation and iterations) of the hybrid identification algorithm. It also shows the preferred network architectures.

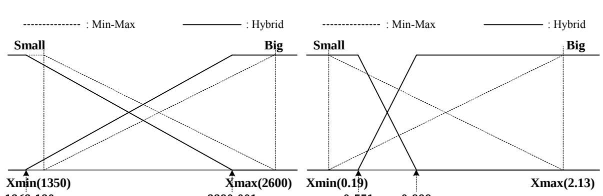
Fig. 12 shows the optimization process by visualizing learning rate and momentum coefficient in successive cycles (generation and iterations) of the hybrid identification algorithm.

Table 4. Parameters of the optimization environment and computational effort.

Genetic Algorithms		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

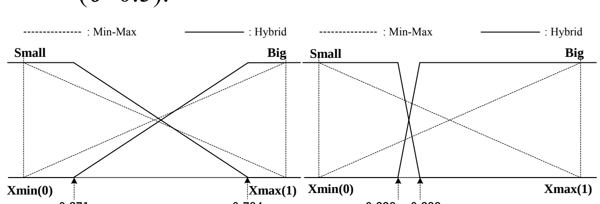
Table 5. Performance index as a function of the weighting factor.

θ	Simplified inference method		Linear inference method	
	PI	E_PI	PI	E_PI
0.0	13.695	11.960	10.753	12.010
0.25	13.274	11.997	10.708	12.018
0.5	12.943	12.176	10.584	12.108
0.75	12.795	12.425	10.185	12.968
1.0	12.617	13.890	8.322	30.561



(a-1) MLSS (a-2) WSR

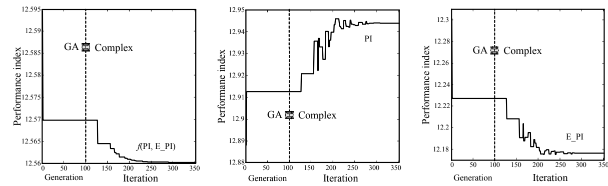
(a) Simplified inference method-based FRFNN (θ=0.5).



(b-1) MLSS (b-2) WSR

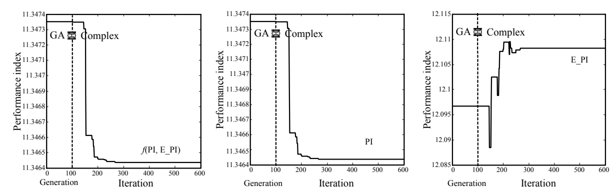
(b) Linear inference method-based FRFNN (θ=0.5).

Fig. 10. The final tuned values of membership functions by hybrid identification algorithm.



(a-1) $f(PI, E\_PI)$ (a-2) PI (a-3) E_PI

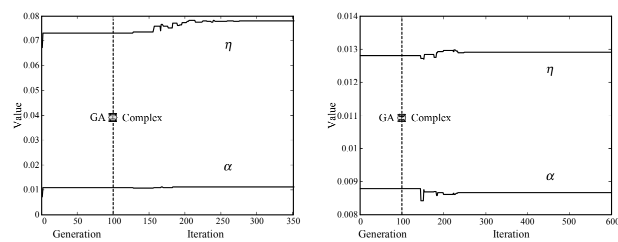
(a) Simplified inference method-based FRFNN (θ=0.5).



(b-1) $f(PI, E\_PI)$ (b-2) PI (b-3) E_PI

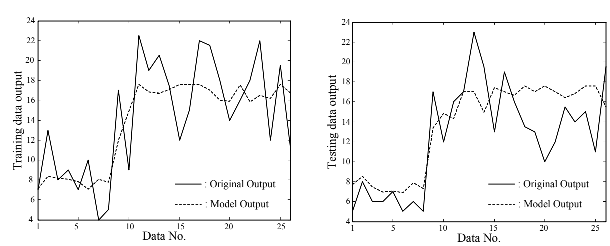
(b) Linear inference method-based FRFNN (θ=0.5).

Fig. 11. The optimization process of each performance indexes by the hybrid identification algorithm.

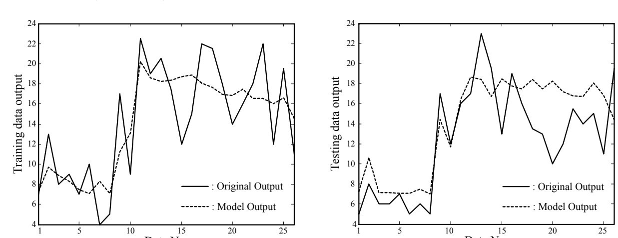


(a) Simplified (b) Linear

Fig. 12. The search process to optimal parameters by hybrid identification algorithm (θ=0.5).



(a) Simplified inference method-based FRFNN (θ=0.5).



(b) Linear inference method-based FRFNN (θ=0.5).

Fig. 13. Original output and model output of relation-based fuzzy-neural networks.

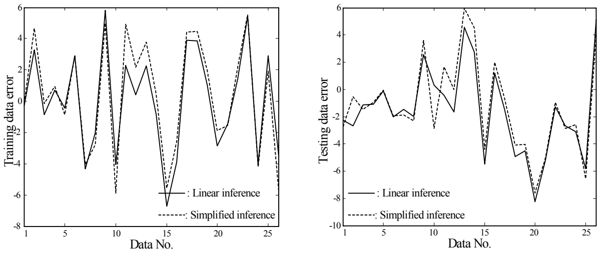


Fig. 14. Errors curves of relation-based fuzzy-neural networks.

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

MODEL			PI_s	E_PI_s
Fuzzy model [7]	Simplified		13.72	16.20
	Linear		6.39	54.23
Hybrid fuzzy model [8]	Simplified	θ=0.5	12.403	12.200
	Linear	θ=0.5	7.175	24.658
Fuzzy set-based FNN model [11]	Simplified	θ=0.6	13.40	8.28
	Linear	θ=0.5	12.30	9.82
Our model	Simplified	θ=0.5	12.943	12.176
	Linear	θ=0.5	10.584	12.108

The original output and model output are shown in Fig. 13. The errors of relation-based fuzzy-neural networks are shown in Fig. 14.

5. CONCLUSIONS

In this paper, the hybrid identification algorithm is presented to automatically extract the optimal parameters of the Fuzzy Relation-based Fuzzy-Neural Networks (FRFNN) from complex nonlinear datasets. The main contributions of this paper are as follows: 1) The hybrid Identification algorithm is used for auto-tuning of the parameters of FRFNN model such as apexes of the membership functions, learning rates, and momentum coefficients. 2) The hybrid identification algorithm combines GAs with the improved complex method to guarantee both global optimization and local convergence. 3) 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 sewage treatment process. The experimental studies clearly revealed that we could obtain better performance (both approximation and generalization capabilities) for two commonly used experimental datasets.

REFERENCES

[1] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, pp. 801-806, 1992.

[2] Y. C. Chen and C. C. Teng, "A model reference control structure using a fuzzy neural networks," *Fuzzy Sets and Systems*, vol. 73, pp. 291-312, 1995.

[3] J. Zhang and A. J. Morris, "Fuzzy neural networks for nonlinear systems modeling," *IEE Proc.-Control Theory and Application*, vol. 142, pp. 551-556, 1995.

[4] D. E. Golderg, *Genetic Algorithm in Search: Optimization & Machine Learning,* Addison Wesley*,* 1989.

[5] 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.

[6] 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.

[7] 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 Systems*, vol. 115, no. 2, pp. 205-230, 2000.

[8] 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, January. 2002.

[9] 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.

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

[11] S.-K. Oh, W. Pedryc, and H.-S. Park, "Hybrid identification in fuzzy-neural networks," *Fuzzy Sets & Systems*, 2003 (in press).

[12] S.-K. Oh, W. Pedrycz, and H.-S. Park, "Multi-FNN identification by means of hcm clustering and genetic algorithms," *Fuzzy Sets and Systems*, 2002.

[13] S.-K. Oh, W. Pedrycz , and H. S. Park, "Implicit rule-based fuzzy-neural networks using the identification algorithm of GA hybrid scheme based on information granulation," *Advanced Engineering Informatics,* 2003 (in press).

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

[15] S.-K. Oh, *Computational Intelligence by Pro-*

gramming focused on Fuzzy, Neural Networks, and Genetic Algorithms, Naeha Press, 2002.

[16] Ho-Sung Park and Sung-Kwun Oh, "Multi-FNN identification based on hcm clustering and evolutionary fuzzy granulation," *International Journal of Control, Automation, and Systems*, vol. 1, no. 2, pp 194-202, June 2003.

[17] Ho-Sung Park and Sung-Kwun Oh, "Rule-based FNN using the identification algorithm of GA hybrid scheme," *International Journal of Control, Automation, and Systems*, vol. 1, no. 1, pp 101-110, March 2003.

Ho-Sung Park received his B.S. and M.S. degrees in Control and Instrumentation Engineering from 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 KIEE and ICASE.

Sung-Kwun Oh received his 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 for the Korean Institute of Electrical Engineers (KIEE) and the Institute of Control, Automation & Systems Engineers (ICASE), Korea.