A Structure Identification Method of Submodels for Hierarchical Fuzzy Modeling Using the Multiple Objective Genetic Algorithm

Kanta Tachibana,[1,*] Takeshi Furuhashi[2,†]
[1] *Complex System Engineering Laboratory, Department of Computational Science and Engineering, Graduate School of Engineering, Nagoya University, Furou-cho, Chikusa-ku, Nagoya 464-8603, Japan*
[2] *Department of Information Engineering, Mie University, 1515 Kamihama-chor, Tsu 514-8507, Japan*

Fuzzy models describe nonlinear input-output relationships with linguistic fuzzy rules. A hierarchical fuzzy modeling is promising for identification of fuzzy models of target systems that have many input variables. In the identification, (1) determination of a hierarchical structure of submodels, (2) selection of input variables of each submodel, (3) division of input and output space, (4) tuning of membership functions, and (5) determination of fuzzy inference method are carried out. This article presents a hierarchical fuzzy modeling method with an uneven division method of input space of each submodel. For selecting input variables of submodels, the multiple objective genetic algorithm (MOGA) is utilized. MOGA finds multiple models with different input variables and different numbers of fuzzy rules as compromising solutions. A human designer can choose desirable ones from these candidates. The proposed method is applied to acquisition of fuzzy rules from cyclists' pedaling data. In spite of a small number of data, the obtained model was able to give detailed suggestions to each cyclist. © 2002 Wiley Periodicals, Inc.

1. INTRODUCTION

Fuzzy modeling[1,2] is a method to describe nonlinear input-output relationships using fuzzy rules. For automatic acquisition of fuzzy rules, combinations of fuzzy logic and neural networks have been studied.[3–11] The fuzzy neural network (FNN) in Refs. 5 and 9 is capable of identifying fuzzy rules and tuning the membership functions by means of backpropagation learning. This FNN has been applied to the fuzzy modeling of nonlinear systems.[10]

*e-mail: kanta@cmplx.cse.nagoya-u.ac.jp.
†Author to whom all correspondence should be addressed; e-mail: furuhashi@pa.info.mie-u.ac.jp.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 17, 495–513 (2002)

Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/int.10034

Division of input space of fuzzy models is one of the important steps of fuzzy modeling. This division is a coarse setting of model parameters, thus it determines the performance of fine tuning using an FNN. For an appropriate division of input space, several methods have been reported.[12–16] These conventional methods have achieved the division by merging similar membership functions or inserting new ones. This article proposes a new method for dividing the input space unevenly based on model errors.

As it is hard to obtain a sufficient data set from an actual plant with many input variables that covers the whole input space, a hierarchical fuzzy modeling method using multiple FNNs[11] was proposed. Each submodel in the hierarchical fuzzy model has a smaller number of input variables. Therefore, it does not need many data for the description of the input-output relationships in the subspace.

Karr, Freeman, and Meredith[17] proposed a combination of fuzzy logic and a genetic algorithm (GA). A GA finds fuzzy rules using the payoff for the success/failure of its actions. A GA was applied to identification of the hierarchical structure of the fuzzy model[18] from given input-output pairs of data. Matsushita, et al.[19] and Furuhashi, et al.[20] have applied a GA to selection of input variables in hierarchical models. This method is very effective in the case where the plant has a strong nonlinearity.

This article studies the performance of hierarchical fuzzy modeling using the multiple objective genetic algorithm (MOGA).[22] Two types of coding are tested for acquiring various fuzzy models on a Pareto front. Numerical experiments using a data set generated from a nonlinear equation are done. The results show that a new coding that decides the number of divisions of input space of each submodel is effective for generating various models on the Pareto front. The proposed division method and the new coding method are applied to a fuzzy modeling of 14 cyclists' pedaling. It is shown that useful suggestions for each cyclist were extracted from the obtained model.

2. FUZZY MODELING

Fuzzy modeling is a method to identify input-output relationships using fuzzy if-then rules. Hierarchical fuzzy modeling is used to carry out the following:

(1) Determination of a hierarchical structure of submodels
(2) Selection of input variables of each submodel from candidates obtained from the modeling object
(3) Division of input and output space of each submodel
(4) Tuning of membership functions
(5) Determination of a fuzzy inference method

In this section, step 3 is mainly discussed. For the division of input space, this section presents a new method based on model errors and a statistical test.

Steps 1 and 2 are discussed in Section 3. For steps 4 and 5, this article uses the fuzzy neural network (FNN) described in Refs. 5 and 9. This FNN is described in the following subsections.

2.1. Fuzzy Rules and a Simplified Fuzzy Inference Method

Suppose that the target system has M inputs and N outputs, and the fuzzy modeling is to identify the input-output relationships with fuzzy rules. The ith rule R^i $(i = 1, \ldots, NOR)$ is described as:

$$R^i: \text{IF } \boldsymbol{x} \text{ is } \boldsymbol{A}^i \quad \text{THEN } \boldsymbol{y} \text{ is } \boldsymbol{B}^i \tag{1}$$

$$\begin{gathered} \boldsymbol{x} = (x_1, x_2, \ldots, x_M), \quad \boldsymbol{y} = (y_1, y_2, \ldots, y_N), \\ \boldsymbol{A}^i = \left(A_1^i, A_2^i, \ldots, A_M^i\right), \quad \boldsymbol{B}^i = \left(B_1^i, B_2^i, \ldots, B_N^i\right) \end{gathered} \tag{2}$$

where the input variables x_m $(m = 1, \ldots, M)$ and the output variables y_n $(n = 1, \ldots, N)$ are real numbers, A_m^i is the fuzzy variable for the mth input variable, and B_n^i is the fuzzy variable for the nth output variable.

This article uses the simplified fuzzy inference method given by:

$$\mu^i(\boldsymbol{x}) = \prod_{m=1}^{M} A_m^i(x_m) \tag{3}$$

$$\widehat{\mu^i(\boldsymbol{x})} = \frac{\mu^i(\boldsymbol{x})}{\sum_{j=1}^{NOR} \mu^j(\boldsymbol{x})} \tag{4}$$

$$y_n^*(\boldsymbol{x}) = \sum_{i=1}^{NOR} \widehat{\mu^i(\boldsymbol{x})} b_n^i \tag{5}$$

where B_n^i in Equation 2 is a singleton denoted by b_n^i, μ^i is the activation value of R^i, $\widehat{\mu^i}$ is the normalized activation value, and y_n^* is the inferred value.

The parameters to be determined are the division of input space, the shapes of membership functions in the antecedent parts, and the singletons in the consequent parts.

2.2. Fuzzy Neural Network

A fuzzy neural network is a good tool to finely tune the parameters, such as the shapes of membership functions and the singletons in the consequent parts. The FNN presented in Refs. 5 and 9 is a multi-layered backpropagation (BP) model with a specially designed structure for easy extraction of fuzzy rules from a trained one. This article uses Type I of the FNNs in Ref. 9. Figure 1 shows an example of the FNN.

This is a case where the FNN has two inputs, x_1 and x_2, one output y, and three membership functions for each input. The backpropagation (BP) learning algorithm can be applied to modify the connection weights w_c, w_g, and w_b.

Figure 2(a) shows an example of membership functions in the antecedent formed in the (A)–(D) layers. The connection weights, $\boldsymbol{w}_c$ and $\boldsymbol{w}_g$, determine the positions and slopes of the sigmoid functions f in the units in the (C) layer, respectively. The output of the (C) layer, the grade of the membership function A_m^i, is

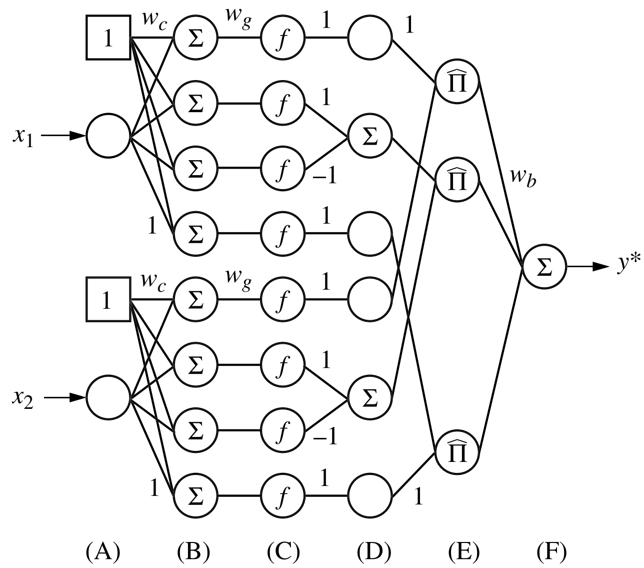


Figure 1. Fuzzy neural network.

given by:

$$A_m^i = f\left(w_g^{i,m} * \left(x_m + w_c^{i,m}\right)\right) \tag{6}$$

Each membership function consists of one or two sigmoid functions. The outputs of the units in the (D) layer are the values of the membership functions. The products of these values are the inputs to the units in the (E) layer. The outputs of the units are the normalized activation values in the antecedent $\widehat{\mu^i}$ in Equation 4. The output of the unit in the (F) layer is the sum of the products of the connection weights $\boldsymbol{w}_b$ and the normalized truth values. The connection weights $\boldsymbol{w}_b$ correspond to the singletons in the consequent parts b_n^i as shown in Figure 2(b). The output in the (F) layer is, therefore, the inferred value y_n^* in Equation 5.

The model infers $\boldsymbol{y}^*(\boldsymbol{x}^s) = (y_1^*(\boldsymbol{x}^s), \ldots, y_N^*(\boldsymbol{x}^s))$ for a teaching signal $(\boldsymbol{x}^s, \boldsymbol{y}^s)$. The error $\boldsymbol{e}^s$ for this teaching signal is given by:

$$\begin{aligned} \boldsymbol{e}^s &= \left(e_1^s, e_2^s, \ldots, e_N^s\right) \\ \forall n\ (n = 1, \ldots, N) \quad e_n^s &= y_n^*(\boldsymbol{x}^s) - y_n^s \end{aligned} \tag{7}$$

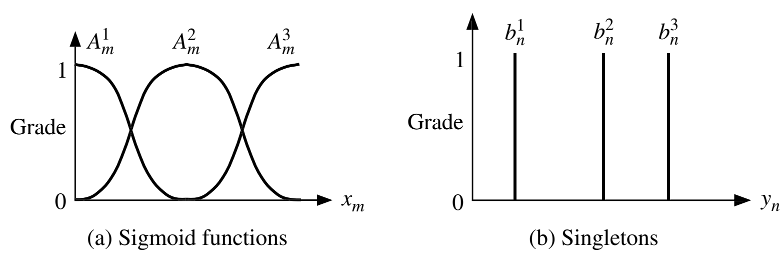


Figure 2. Membership functions.

The parameters of the FNN are updated so that:

$$minimize\ E = \sum_{n}^{N} \sum_{s}^{NOD} \left(e_n^s\right)^2 \tag{8}$$

where *NOD* is the number of sampled data.

Since the center-of-gravity method is used in the (E) layer, the updating method of connection weights, i.e., the BP algorithm, needs some modifications. The learning algorithm for the FNN is well described in Ref. 9.

The feature of this FNN is that fuzzy rules can be extracted easily from the trained FNN. The conventional three-layered neural network can identify the input-output relationships. However, it is hard to extract rules from the trained three-layered neural network.

2.3. Division of Input Space

Initial division of the input space determines the performance of the fine tuning of the FNN. There are three major methods to divide the input space for fuzzy modeling: grid-type, tree-type, and cluster-type. The grid division tends to identify a model with too many rules in the case in which the data set has multiple input variables. In this article, we choose tree division, which divides the input space in a way similar to a decision tree. The number of divisions is made smaller with this division method, and it is easy for us to extract rules from the identified model. The cluster-type division probably makes the simplest model with the smallest number of rules. But the divided subspaces do not cover the input space and the identified model sometimes loses generality. This subsection presents a tree division using the statistical t-test. This test is to examine whether the average value of a group of data is significantly different from that of the other group. This article uses the p-value of the t-test. We apply the t-test as the stopping condition of the following division algorithm:

(1) The I-dimensional input space (we assume that the data have I input variables) is not divided initially; i.e., the number of subspaces equals 1. A fuzzy rule is identified in each subspace; the number of rules, N_R, is the same as the number of subspaces. Thus, N_R is 1 initially. The subspace, which covers the whole input space, is denoted by S_1.

(2) The data are divided into N_R data sets. If the input of a datum is in the rth subspace S_r, this datum belongs to the rth data set. One of subspaces is going to be divided into two. This subspace S_{DIV} is decided as:

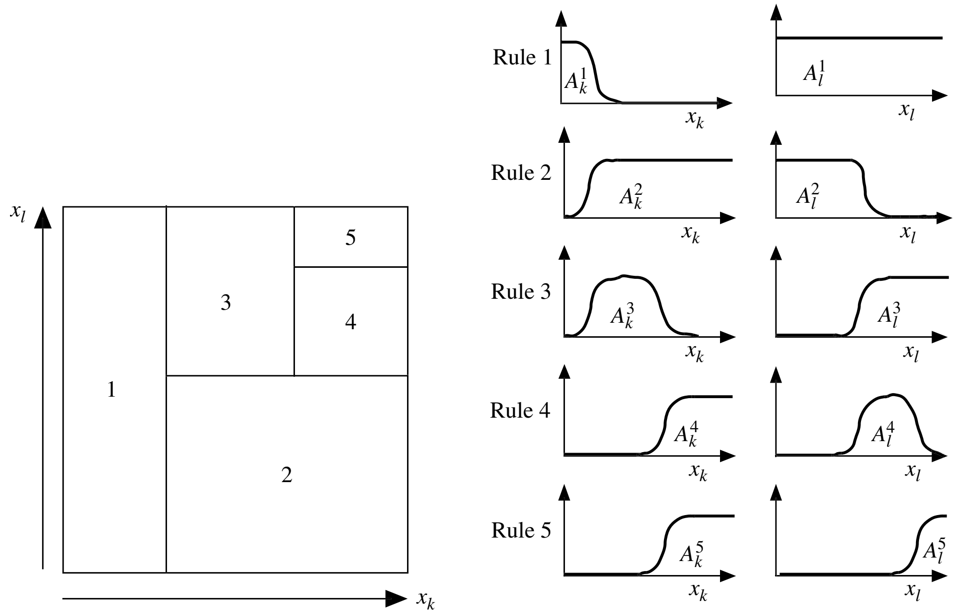
$$S_{DIV} = \arg \max_{r=1,\ldots,N_R} (n_r - 1)V(S_r) \tag{9}$$

where $V(S_r)$ and n_r are the variance of outputs and the number of data in the rth data set, respectively.

The selected subspace for the division is the one with the maximum error. S_{DIV} is divided into new two subspaces, S_{new1} and S_{new2}. The dividing point, x_d, is decided as:

$$\min_{i=1,\ldots,I} \min_{xd_i \in S_{DIV}} \{(n_{new1} - 1)V(S_{new1}(i, xd_i)) + (n_{new2} - 1)V(S_{new2}(i, xd_i))\} \tag{10}$$

where $S_{new1}(i, xd_i p)$ and $S_{new2}(i, xd_i)$ denote the data sets generated by dividing S_{DIV} at xd_i, which is the dividing point on the ith axis. N_R is increased by 1; i.e. $N_R := N_R + 1$.



(a) Uneven division (b) Membership functions

Figure 3. Obtained division and membership functions.

(3) The division of the data set is evaluated by a t-test. If the p-value is less than the preset significance level, step 2 is repeated to divide another subspace. We consider that if the p-value of the t-test to a division is less than a pre-specified level, the average output values in the two divided subspaces are significantly different from each other.
(4) An FNN with N_R rules is generated and trained. The training of the FNN is only to adjust the singletons in the consequent parts for an efficient training.

The division process is repeated until the p-value becomes greater than the significance level, or the number of rules becomes greater than a decided-upon number. Figure 3 shows an example of uneven division of the input space x_k and x_l, and the membership functions.

3. HIERARCHICAL FUZZY MODELING

A hierarchical fuzzy model is effective in the case in which (1) the modeling target has many input variables, and (2) a small amount of data is given.

A fuzzy model with too many rules (empirically one-fifth or more of the number of data) lacks the generality and the estimation ability. For a small-sized data set of 20 input-output pairs, for example, a model with three or four rules tends to have good generality and good estimation value. These three or four rules are all the information extracted from the data. More fuzzy rules can be obtained from the same data set with our hierarchical fuzzy modeling method, by identifying a hierarchical structure of submodels. The model is, of course, effective for a large-sized data set.

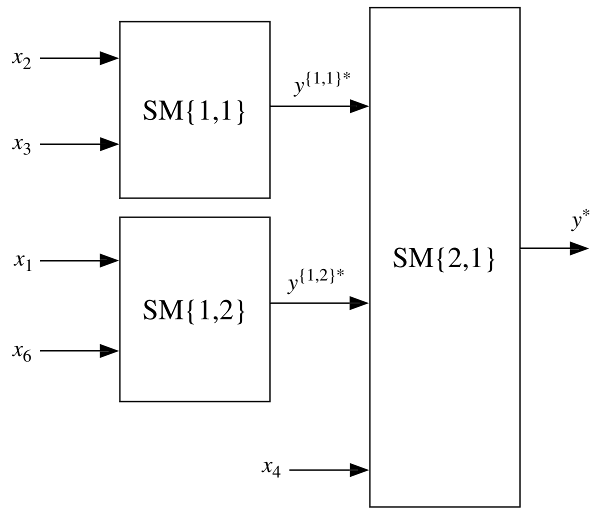


Figure 4. A hierarchical fuzzy model using FNNs.

3.1. Structure of Hierarchical Fuzzy Model

The structure of a hierarchical fuzzy model is illustrated in this subsection. Figure 4 shows an example of a hierarchical fuzzy model, which consists of three FNN submodels. The figure shows a case where the model has a five-dimensional input vector $\boldsymbol{X} = (x_1, x_2, x_3, x_4, x_6)$, a single output $\boldsymbol{y} = (y)$, and a two-layered hierarchical structure. In Figure 4, $y^{\{1,1\}*}$, $y^{\{1,2\}*}$, and y^* are the inferred values of the fuzzy submodels. In the first layer, the submodel SM$\{1,1\}$ is the fuzzy model with the inputs x_2 and x_3. The other fuzzy model with x_1 and x_6 are lined up in parallel. The outputs of these models are $y^{\{1,1\}*}$ and $y^{\{1,2\}*}$, respectively. These two fuzzy submodels in the first layer greatly contribute to the input-output relationships of the system. The input variable x_4 is used by the fuzzy submodel in the second layer. This input is considered to be used for a small adjustment of the model. This fuzzy model reduces the number of input variables for each submodel. This makes each fuzzy rule simple, and makes the number of fuzzy rules small. This reduction of fuzzy rules prevents the model from over-fitting. The obtained fuzzy model, therefore, has generality.

The authors have proposed a hierarchical fuzzy modeling method using FNNs and a GA.[11,19] Each submodel was built using an FNN. A proper set of input variables and sets of membership functions for the submodel were selected by a GA. A hierarchical structure was constructed by finding proper submodels one by one.

3.2. Procedure of Hierarchical Fuzzy Modeling

This subsection describes the procedure of constructing a hierarchical fuzzy model. In this subsection, the modeled system is assumed to have an M-dimensional input vector $\boldsymbol{X}$ and a single output y.

(1) The number of layer h and the model number in the layer j are initially set to 1; i.e., $h := 1$ and $j := 1$.

(2) In this step, the structure of the jth submodel in the hth layer is identified. The submodel is denoted by SM$\{h, j\}$. This model is identified using the method in Section 3.3. A set of input variables are selected out of the candidates of inputs $\boldsymbol{X}$ by the multiple objective genetic algorithm.

If j equals 1, this submodel is the first one in the hth layer to be identified. For other models, the errors of SM$\{h, j-1\}$ are used. The teaching signal $y^{\{h,j\}}$ is given by:

$$y^{\{h,j\}} := \begin{cases} y & (j = 1) \\ e^{\{h,j-1\}} & (j \neq 1) \end{cases}$$

If the sum of outputs of the submodels in the previous layer $(h-1)$ is better than that of the obtained fuzzy model SM$\{h, 1\}$, the sum is used as the final inference value, and this process is stopped. If not, go to step 3.

(3) The next step is the fine tuning of the acquired model SM$\{h, j\}$ by the BP learning of the FNN.[9] The membership functions in the antecedent parts and the singletons in the consequent parts are modified to obtain the best model.

(4) j is increased by 1 ($j := j + 1$). If j is smaller than a preset number J (maximum number of models allowed in a layer), go to step 2. If j equals J, submodels in the next layer are to be identified. The outputs of submodels in the current layer can also be used as input variables of submodels in the next layer:

$$\forall j\ (j = 1, \ldots, J) \quad X_{M+j} := y^{\{h,j\}*} \tag{11}$$

$$M := M + J \tag{12}$$

where $y^{\{h,j\}}$ denotes the output of SM$\{h, j\}$. Set $j := 1$, $h := h + 1$, and go to step 2.

3.3. Identification Methods of Submodel

This subsection introduces two methods of structure identification of each submodel. Both methods utilize the multiple objective genetic algorithm (MOGA).

The first method encodes the combination of input variables. Figure 5 shows an example of the chromosome. This method is called method 1 in this work.

If a gene contains 1, the corresponding variable is used as an input variable of the submodel. If it contains 0, the variable is not used.

The other method encodes the maximum allowable number of fuzzy rules as well as the combination of input variables. Figure 6 shows an example of the chromosome. The number of fuzzy rules is encoded in n_1 to n_K by binary numbers. In this case, the stopping condition for the division of input space is not the preset

x_1	x_2	x_3	x_4	...	x_M
1	0	1	0	...	1

Figure 5. An example of the chromosome for method 1.

x_1	x_2	...	x_M	n_1	...	n_k
1	0	...	0	1	...	1

Figure 6. An example of the chromosome for method 2.

p-value. The input space determined by the chromosome is divided up into the number given by the chromosome.

$\boldsymbol{g}(p) = (g_1(p), g_2(p), \ldots, g_L(p))$ denotes the pth chromosome in the population. Each gene has a binary number, 0 or 1, as shown in the example. The length of each chromosome L is the same as M in the first method and $M + K$ in the second method. The population, that is the number of chromosomes, is denoted by P.

The evolution of individuals is carried out by the following procedure:

(1) Chromosomes are initialized to contain 0 in each gene. The binary number in each gene is flipped to 1 with the probability of pr_{ini}.

$$\forall p = 1, \ldots, P, \quad \forall n = 1, \ldots, L, \quad g_n(p) = \begin{cases} 1, & \text{with the probability of } pr_{ini} \\ 0, & \text{otherwise} \end{cases}$$

(2) Chromosomes are evaluated. Chromosome $\boldsymbol{g}(p)$ decides a combination of input variables and a number of fuzzy rules. A model θ^p is identified using only the selected input variables and the output variable. The division process of the input space described in Section 2.3 is carried out.

The fitness vector $\boldsymbol{f}(p) = (f_1(p), f_2(p), f_3(p))$ is used for the evaluation of the chromosome. The elements in the fitness vector evaluate estimation ability of the submodel, the number of rules, and the number of input variables, respectively. $f_1(p)$ is Akaike information criterion (AIC)[21] given by:

$$f_1 = N_D \log\left(\frac{E}{N_D}\right) + 2DOF \tag{13}$$

where N_D is the number of data examples, E is the total error in Equation 8 and DOF is the degree of freedom of the model. DOF is given by:

$$DOF = 2N_R - 1 \tag{14}$$

for this uneven division of input space, where N_R is the number of rules. AIC is a good criterion for the estimation capability of submodels. $f_2(p)$ and $f_3(p)$ are the number of rules and the number of selected input variables of the model θ^p, respectively.

$$f_2 = N_R \tag{15}$$

$$f_3 = N_V \tag{16}$$

where N_V is the number of selected input variables. The smaller these values are, the better the submodel is. The solution has to be a trade-off between these values, especially the pairs (f_1, f_2) and (f_1, f_3).

The rank of chromosome $\boldsymbol{g}(p)$ in the population is given by:

$$\text{rank}(p) := 1 + n(p) \tag{17}$$

where $n(p)$ is the number of chromosomes that are "superior" to the chromosome $\boldsymbol{g}^p$. The "superiority" is defined such that $\boldsymbol{g}(q)$ is *superior* to $\boldsymbol{g}(p)$ in the case where:

$$(\forall i = 1, 2, 3) \quad f_i(q) \le f_i(p) \wedge (\exists i = 1, 2, 3) \quad f_i(q) < f_i(p) \tag{18}$$

(3) If rank(p) is 1, the chromosome is elite. Elite chromosomes are reserved for the next generation.

(4) The crossover operation is applied to the whole population except for the elites at the rate of pr_{cr}. Parents are selected according to the total fitness value of chromosomes. The total fitness value of chromosome $\boldsymbol{g}(p)$ is given by 1/rank(p).

The mutation operation is applied to each gene of all the chromosomes except for that of the elite chromosomes at the rate of pr_{mut}.

(5) Stop if the number of elite chromosomes becomes larger than a preset number, or generations become larger than a pre-decided limit number. Otherwise, go to step 2.

Multiple elite models are obtained by the above processes. A human designer should select a desirable model out of the obtained elite models. If the designer regards the conciseness of the model to be more important than the estimation ability, he/she would choose a model with a smaller number of rules.

4. NUMERICAL EXPERIMENT

In this section, two coding methods introduced in Section 3.3 are examined. Then the proposed hierarchical fuzzy modeling method is applied to acquisition of fuzzy rules from a data set of 14 cyclists.[26]

4.1. Effect of Coding Method in Section 3.3

We used the following nonlinear equation as the modeling target:

$$y = \sin(\pi x_1) + \cos(\pi(x_1 + x_2)) + \cos(\pi x_1 x_3) + \sin(\pi(x_1 - x_4)) \qquad (19)$$

A variable x_5 that has no relationship with y was also used. The value of each input was set at {0, 0.5, 1}. ($3^5 =$) 243 data examples were generated from this equation.

The first fuzzy submodel in the first layer, $FM^{1,1}$, was identified. Figure 7 shows the value of AIC versus the numbers of rules of all the possible combinations of less than four input variables.

Figure 8 shows an example of elite models obtained by method 1. Seven solutions were obtained. Figure 9 shows another example of candidate models obtained

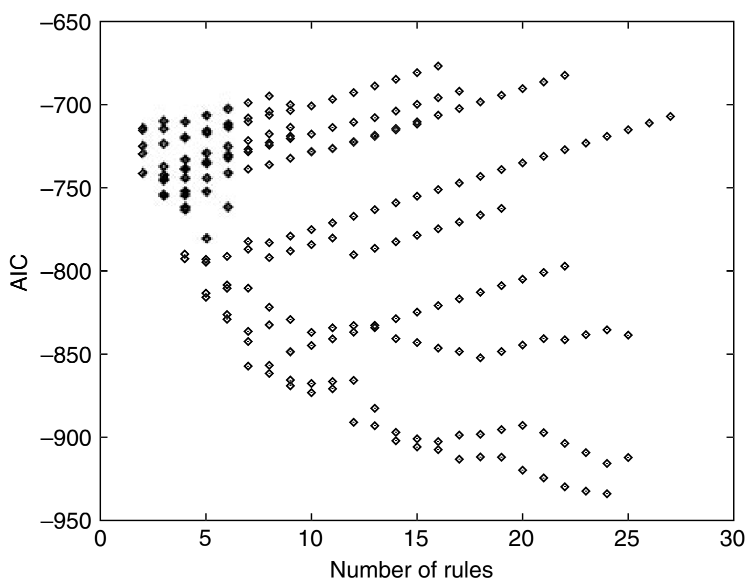


Figure 7. All possible solutions with less than four inputs.

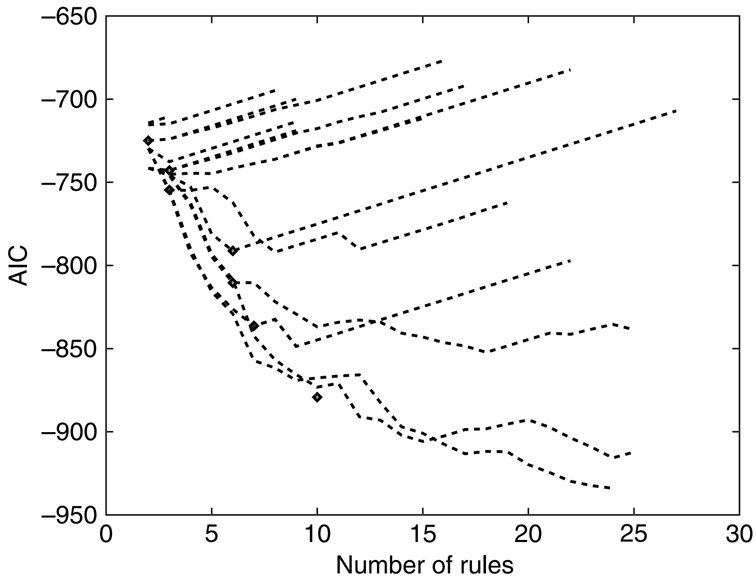


Figure 8. Models obtained by method 1.

by method 2. Fifteen solutions were obtained. These models generated by method 2 were spread widely over the Pareto front. The significant difference in these results was due to the difference in the coding method. Recall that method 2 let the number of rules N_R be determined by a GA. And the MOGA with method 2 could generate various candidates.

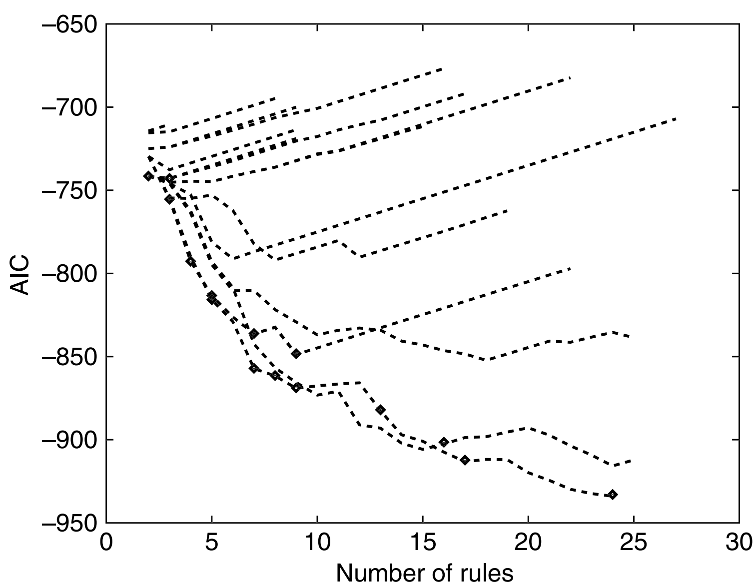


Figure 9. Models obtained by method 2.

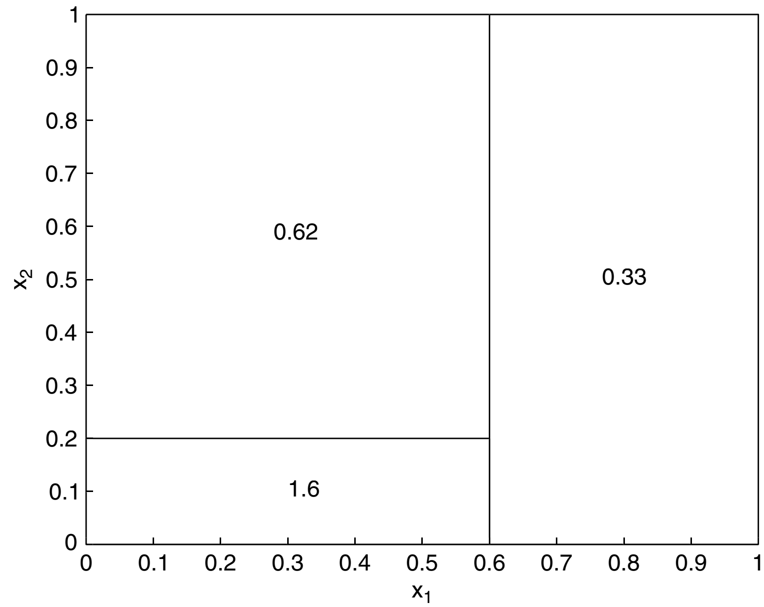


Figure 10. Obtained division of input space with three rules.

Figure 10 and Figure 11 show two examples of the obtained divisions of input space. In both cases, x_1 and x_2 were selected as input variables. The obtained fuzzy rules are shown in Table I and Table II, respectively. The fuzzy rules can be expressed by linguistic labels as shown in the tables.

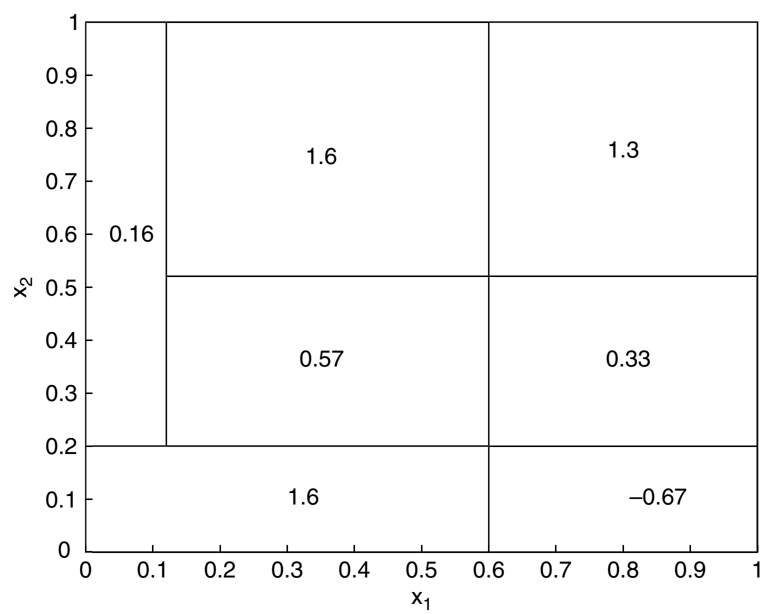


Figure 11. Obtained division of input space with seven rules.

Table I. Obtained rule set with three fuzzy rules.

IF	x_1 is not large and	x_2 is small	THEN $y = 1.6$
IF	x_1 is large		THEN $y = 0.33$
IF	x_1 is not large and	x_2 is not small	THEN $y = 0.62$

Table II. Obtained rule set with seven fuzzy rules.

IF	x_1 is not large	and	x_2 is small	THEN $y = 1.6$
IF	x_1 is large	and	x_2 is small	THEN $y = -0.67$
IF	x_1 is small	and	x_2 is not small	THEN $y = 0.16$
IF	x_1 is medium	and	x_2 is medium	THEN $y = 0.57$
IF	x_1 is large	and	x_2 is medium	THEN $y = 0.33$
IF	x_1 is medium	and	x_2 is large	THEN $y = 1.6$
IF	x_1 is large	and	x_2 is large	THEN $y = 1.3$

Table III. Obtained rule set with 21 fuzzy rules.

IF	x_1 is small	and	x_2 is not large	and	x_3 is small	THEN $y = 0.68$
IF	x_1 is large	and	x_2 is small	and	x_3 is small	THEN $y = 0.20$
IF	x_1 is small	and	x_2 is small	and	x_3 is not small	THEN $y = 0.70$
IF	x_1 is large	and	x_2 is medium	and	x_3 is small	THEN $y = 0.40$
IF	x_1 is small	and	x_2 is medium	and	x_3 is not small	THEN $y = 0.50$
IF	x_1 is large	and	x_2 is large	and	x_3 is small	THEN $y = 0.60$
IF	x_1 is medium	and	x_2 is small	and	x_3 is small	THEN $y = 0.91$
IF	x_1 is medium	and	x_2 is medium	and	x_3 is not small	THEN $y = 0.41$
IF	x_1 is medium	and	x_2 is large	and	x_3 is medium	THEN $y = 0.71$
IF	x_1 is large	and	x_2 is small	and	x_3 is medium	THEN $y = 0.40$
IF	x_1 is large	and	x_2 is medium	and	x_3 is medium	THEN $y = 0.60$
IF	x_1 is large	and	x_2 is large	and	x_3 is large	THEN $y = 0.60$
IF	x_1 is small	and	x_2 is large	and	x_3 is small	THEN $y = 0.40$
IF	x_1 is small	and	x_2 is medium	and	x_3 is not small	THEN $y = 0.30$
IF	x_1 is large	and	x_2 is medium	and	x_3 is large	THEN $y = 0.40$
IF	x_1 is large	and	x_2 is medium	and	x_3 is large	THEN $y = 0.80$
IF	x_1 is large	and	x_2 is small	and	x_3 is large	THEN $y = 0.20$
IF	x_1 is medium	and	x_2 is small	and	x_3 is medium	THEN $y = 0.71$
IF	x_1 is medium	and	x_2 is not small	and	x_3 is small	THEN $y = 0.81$
IF	x_1 is medium	and	x_2 is large	and	x_3 is large	THEN $y = 0.51$
IF	x_1 is medium	and	x_2 is small	and	x_3 is large	THEN $y = 0.51$

Table III shows another set of fuzzy rules obtained by method 2. There were 21 fuzzy rules. These rules had three input variables. This model had less amount of error, but the number of rules was large. Human users can choose one of the solutions according to his/her own purpose.

4.2. Analysis of Cyclists' Pedaling Technique

The proposed hierarchical fuzzy modeling method with the tree-division method and the coding method 2 was applied to extraction of pedaling know-how from pedaling data of 14 cyclists.[26] All cyclists were measured while they were

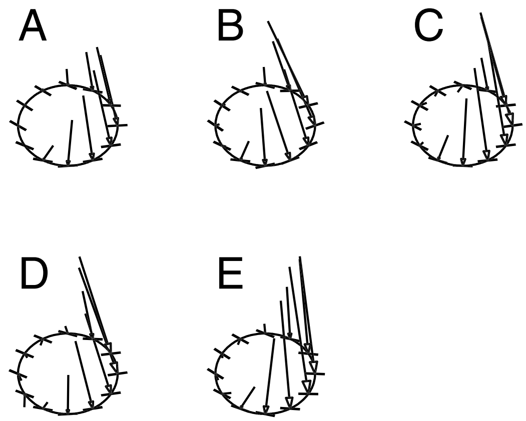


Figure 12. Force patterns of five cyclists.

pedaling at their maximal effort. Six cyclists of the 14 were also measured while they were pedaling at 90 percent of their maximal outputs. The number of data was 20. The data set consisted of pedal angle θ, and normal and tangential forces to the pedal as functions of crank angle ranged from 0 to 360 degrees. Two variables, absolute value of the force exerted on the pedal, $|F|$, and angle of the force direction to the pedal, θ_F, were calculated. The variables $|F|, \theta_F$, and θ were averaged in every 90 degrees of the crank angle. Three variables in four ranges, for a total of 12 variables, were obtained as the inputs. The output variable y is the average output power [watt]. Figure 12 depicts force patterns of five cyclists. Cyclist A had the weakest output among the 14. Cyclist E was the best of the 14. Common interests by cyclists with this kind of data were: the factor that affected the output most; and the point that they should concentrate on to improve their outputs.

4.2.1. Hierarchical Fuzzy Model of Cyclists' Data

Hierarchical fuzzy modeling was utilized to analyze the data. Figure 13 depicts the final structure of the obtained hierarchical model. The submodels SM{1-1} and SM{1-2} were identified. The outputs of the two obtained submodels were added in the second layer.

Figure 14(a) shows the solutions for the first submodel in the first layer searched by the MOGA. The solutions, which were on the Pareto front in the final generation, are indicated by circles. We selected the model with four rules and two inputs out of the compromising solutions as the submodel SM{1, 1}. Table IV shows the fuzzy rules of this submodel. The two inputs were $|F|_1$, the force strength, and θ_{F1}, the force direction, in the first half of downstroke. Linguistic labels and ranges in which the grades are greater than 0.5 are shown. The consequent singletons, y_1, are also shown. The universe of discourse of the input is shown in the bottom row. The rules were interesting. Though the force $|F|_1$ of rule 3 was stronger than that of rule 2, the application of the force was to a wrong direction; i.e., a small θ_{F1} limited the output.

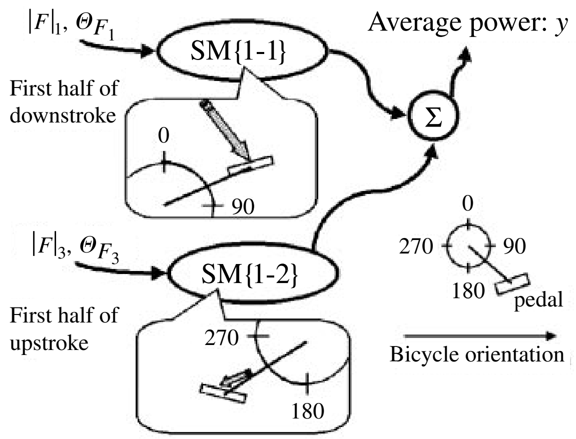


Figure 13. Obtained hierarchical model.

Further information would not be obtained from this quite small amount of data unless hierarchical modeling was utilized.

The second submodel SM{1, 2} was identified using the error of submodel SM{1-1} as its output. Figure 14(b) shows the solutions for submodel SM{1-2}.

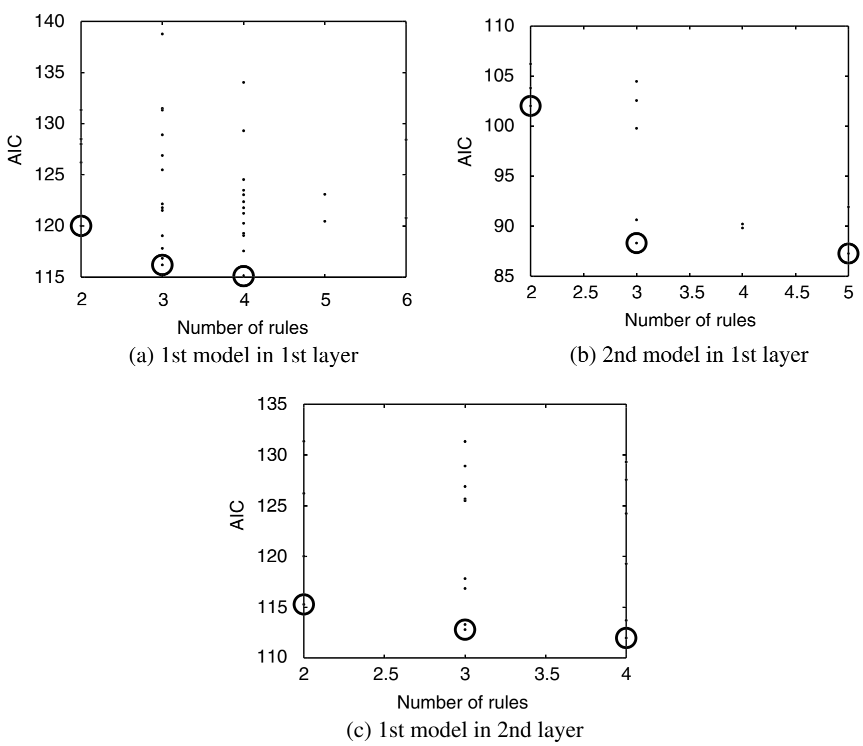


Figure 14. Solutions searched by MOGA for each submodel.

Table IV. Fuzzy rules in submodel SM{1-1}.

No	$\vert F\vert_1$	θ_{F1}	y_1
1	Large (315∼)	Large (−78∼)	253
2	Medium (274 ∼ 315)	—	209
3	Large (315∼)	Small (∼−78)	207
4	Small (∼274)	—	184
	176 ∼ 371	−83 ∼ −48	

Table V. Fuzzy rules in submodel SM{1-2}.

No	$\vert F\vert_3$	θ_{F3}	y_2
1	Large (97∼)	Large (−137∼)	+16
2	—	Small (∼−137)	+6
3	Small (∼97)	Large (−137∼)	−15
	42 ∼ 114	−244∼−97	

We chose the model with three rules and two inputs. Table V shows the rules. The strength $|F|_3$ and the force direction θ_{F3} in the first half of upstroke were selected as the inputs of the compensative submodel. These rules were also interesting. If the force direction was small (nearly parallel to the pedal surface), the strength did not contribute to the output. If it was large, the error of SM{1, 1} depends on $|F|_3$.

For the first submodel in the second layer, SM{2,1}, the outputs of submodels in the first layer were also the candidates of its inputs. Figure 14(c) shows the solutions for submodel SM{2-1}. We chose the model with four rules and two inputs. Table VI shows the rules. The rules indicate that the compensation by the submodel SM{1-2} was needed only for the cyclists in intermediate levels. But, the model did not have less error than the sum of the outputs of submodel SM{1-1} and submodel SM{1-2}. So, in the second layer, a sum unit was employed instead of the fuzzy model SM{2, 1}.

The root-mean-squared error of the hierarchical fuzzy model was 7.1 watts, which was 3.3 percent of the average output, 211 watts. The model had 12 degrees of freedom in total. A linear regression with 13 degrees of freedom gave a smaller error. The root-mean-squared error with the linear regression was 4.8 watts. However, more detailed information was extracted from the fuzzy model as discussed in the next subsection.

Table VI. Fuzzy rules in submodel SM{2-1}.

No	y_1	y_2	y
1	Large (229∼)	—	253
2	Medium (196 ∼ 229)	Large (−15∼)	212
3	Medium (196 ∼ 229)	Small (∼−15)	188
4	Small (∼196)	—	184
	183 ∼ 266	−18 ∼ +12	

Table VII. Activation values of fuzzy rules of five cyclists.

	Rule no (consequent singleton)							Output	
	1 (253)	2 (209)	3 (207)	4 (184)	1 (+16)	2 (+6)	3 (−15)	Inferred	Actual
A	0	0	0	100	0	51	49	176.2	163.0
B	29	61	8	2	3	33	64	203.5	196.4
C	11	0	74	16	70	9	20	222.7	230.5
D	61	37	1	1	0	100	0	244.2	248.2
E	97	0	1	2	63	28	9	278.5	279.6

4.2.2. Discussion

The fuzzy model gave us more information about the pedaling technique. The strength and the force direction in the first half of downstroke were important. And those in the first half of upstroke were also important. This information was difficult to extract from the linear regression. The correlation coefficients of input variables to the output were 0.78 at the maximum. The most related input variable to the output was $|F|_2$: strength in the second half of downstroke. The second maximal coefficient was 0.60 of the strength in the first half of downstroke, $|F|_1$. The correlation coefficients of the force directions in the halves of downstroke θ_{F1} and θ_{F2} were −0.08 and 0.47, respectively. The correlation coefficients of the force strength $|F|_3$ and direction θ_{F3} in the first half of upstroke, which were selected by the fuzzy model, were 0.52 and 0.20, respectively.

The fuzzy model found out the combination effects of the force strength and the direction, which the linear regression could not reveal. Furthermore, the fuzzy model was able to give detailed information as follows: Table VII shows the normalized activation values of four fuzzy rules in SM{1, 1} and three fuzzy rules in SM{1, 2} in the case of five cyclists, A–E. The activation values are normalized so that the sum of all the activation values of a submodel is 100 percent.

Cyclist A matched the fourth rule in submodel SM{1-1}, and matched the second and the third rules in submodel SM{1-2} half and half. Referring to Table IV, the major cause of his relatively low performance was explained by the small force in the first half of downstroke. And referring to Table V, the minor cause was explained by the application of force to a wrong direction in the first half of upstroke.

According to the second row in Table VII, Table IV, and Table V, the pedaling of cyclist B was characterized by the medium force in the first half of downstroke and the small force in the first half of upstroke. He should take more care in his upstroke than downstroke.

The third line in Table VII and the rule tables indicate that the pedaling of cyclist C was characterized by the strong but wrong directional force in the first half of downstroke and the large force in the first half of upstroke. He should mind the force direction in his downstroke.

Cyclist D had acquired proper force direction in the first half of downstroke, because the first and the second rules of submodel SM{1-1} had higher activation values. He should improve strength in downstroke and correct the direction in upstroke.

These pieces of advice were extractable because the acquired fuzzy models had interpretability. Especially for a small data set, the proposed hierarchical fuzzy modeling is effective.

5. CONCLUSION

This article presented a hierarchical fuzzy modeling method. A new division method of input space of each submodel based on model errors was proposed. The MOGA was utilized to determine a combination of input variables and number of rules of each submodel. Two coding methods were proposed and tested. The coding method that encodes both combinations of input variables and number of rules generated various compromising solutions on the Pareto front. A human designer can choose the desirable one from the candidates.

The proposed hierarchical fuzzy modeling method was applied to acquisition of fuzzy rules from cyclists' pedaling data. In spite of a small number of data, the obtained model was able to give detailed suggestions to each cyclist.

References

1. Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybernet 1985;15(1):116–132.
2. Kang G, Sugeno M. Fuzzy modeling. Trans Society Instrument Control Engineers 1987;23(6):650–652.
3. Takagi H, Hayashi I. Artificial-neural-network driven fuzzy reasoning. IIZUKA-88, 1988. pp 183–184.
4. Furuya T, Kokubu A, Sakamoto T. NFS: Neuro fuzzy inference system. IIZUKA-88, 1988. pp 219–230.
5. Horikawa S, Furuhashi T, Okuma S, Uchikawa Y. A fuzzy controller using a neural network and its capability to learn expert's control rules. In: Proc Intl Conf on Fuzzy Logic & Neural Networks (IIZUKA-90), 1990. pp 103–106.
6. Yamaguchi T, Imasaki N, Haruki K. Fuzzy rule realization on associative memory system. IJCNN-90-WASH-DC 1990;II:720–723.
7. Yoshida K, Hayashi Y, Imura A, Shimada N. Fuzzy neural expert system for diagnosing hepatobiliary disorders. In: Proc Intl Conf on Fuzzy Logic & Neural Networks (IIZUKA-90), 1990. pp 539–543.
8. Sanchez E. Fuzzy connectionist expert systems. In: Proc Intl Conf on Fuzzy Logic & Neural Networks (IIZUKA-90), 1990. pp 31–35.
9. Horikawa S, Furuhashi T, Uchikawa Y. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm. IEEE Trans Neural Networks 1992;3(5):801–806.
10. Hasegawa T, Horikawa S, Furuhashi T, Uchikawa Y. A study on fuzzy modeling of BOF using a fuzzy neural network. In: Proc 2nd Intl Conf on Fuzzy Logic & Neural Networks (IIZUKA'92), 1992. pp 1061–1064.
11. Nakayama S, Furuhashi T, Uchikawa Y. A proposal of hierarchical fuzzy modeling method. J Japan Society Fuzzy Theory Syst 1993;5(5):1155–1168.
12. Yager RR, Filev DP. United structure and parameter identification of fuzzy models. IEEE Trans Syst Man Cybernet 1993;23(4):1198–1205.
13. Song BG, Marks II RJ, Oh S, Arabshahi P, Caudel TP, Choi JJ. Adaptive membership function fusion and annihilation in fuzzy if-then rules. In: Proc 2nd IEEE Intl Conf on Fuzzy Systems, 1993. pp 961–967.
14. Chao CT, Chen YJ, Teng CC. Simplification of fuzzy-neural systems using similarity analysis. IEEE Trans Syst Man Cybernet 1996;26(2):344–354.

15. Babuska R, Setnes M, Kaymak U, van Nauta Lemke HR. Rule base simplification with similarity measures. In: Proc 5th Intl Conf on Fuzzy Systems, 1996. pp 1642–1647.
16. Yen J, Wang L. An SVD-based fuzzy model reduction strategy. In: Proc 5th Intl Conf on Fuzzy Systems, 1996. pp 835–841.
17. Karr CL, Freeman L, Meredith D. Improved fuzzy process control of spacecraft autonomous rendezvous using a genetic algorithm. In: SPIE Conf on Intelligent Control and Adaptive Systems, 1989. pp 274–283.
18. Shimojima K, Fukuda T, Hasegawa Y. Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm. Fuzzy Sets Syst 1995;71(3):295–309.
19. Matsushita S, Kuromiya A, Yamaoka M, Furuhashi T, Uchikawa Y. Determination of antecedent structure of fuzzy modeling using genetic algorithm. In: Proc 1996 IEEE Intl Conf on Evolutionary Computation (ICEC'96), 1996. pp 235–238.
20. Furuhashi T, Matsushita S, Tsutsui H. Evolutionary fuzzy modeling using fuzzy neural networks and genetic algorithm. In: Proc 1997 IEEE Intl Conf on Evolutionary Computation (ICEC'97), 1997. pp 623–627.
21. Akaike H. Information theory and an extention of the maximum likelihood principle. 2nd Inter Symp on Information Theory, 1973. pp 267–281.
22. Fonseca CM, Fleming PJ. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: Proc 5th Int Conf on Genetic Algorithms, 1993. pp 416–423.
23. Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In: Proc 1st Intl Conf Genetic Algorithms, 1985. pp 93–100.
24. Ishibuchi H, Nozaki K, Yamamoto N, Tanaka H. Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Trans Fuzzy Syst 1995;3(3):260–270.
25. Ishibuchi H, Murata T, Turksen IB. Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. Fuzzy Sets Syst 1997;89(2): 135–150.
26. Kautz SA, Feltner ME, Coyle EF, Bayler AM. The pedaling technique of elite endurance cyclist: Changes with increasing workload at constant cadence. Int J Sport Biomechanics 1991;7:29–53. (http://isb.ri.ccf.org/data/kautz/)