Automatic Generation of Fuzzy Classification Systems Using Hyper-Cone Membership Functions

**Proceedings 2003 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
July 16-20, 2003, Kobe, Japan**

Automatic Generation of Fuzzy Classification Systems Using Hyper-Cone Membership Functions

Lina HNG
Graduate School of Education
Fukui University
3-9-1 Bunkyo, Fukui, JAPAN 910-8507
linahng@centrin.net.id

Hiroyuki INOUE
Faculty of Education and Regional Studies
Fukui University
3-9-1 Bunkyo, Fukui, JAPAN 910-8507
inoue@edu00.f-edu.fukui-u.ac.jp

Kenji MIYASAKA
Faculty of Education and Regional Studies
Fukui University
3-9-1 Bunkyo, Fukui, JAPAN 910-8507

Mitsuru TSUKAMOTO
Faculty of Education and Regional Studies
Fukui University
3-9-1 Bunkyo, Fukui, JAPAN 910-8507

Abstract

In this paper, we propose automatic generation methods of fuzzy classification rules with the Genetic Algorithms (GAs) to obtain compact fuzzy systems. This time, we propose an approach of hyper-cone membership function to construct rules for the antecedent part. Then, this method is determined the location and shape of hyper-cone membership function in the antecedent part, output class and the number of necessary inputs of each rule by GAs. Also, using the rule addition method in GA process, compact fuzzy classification systems are obtained. Though the proposed methods are quite simple, the process of GAs on both methods presents a solving for two-objective optimization problems: increasing the numbers of correct pattern classification, while decreasing the rule and input numbers optimally. This method was applied to Wine data sets and Wisconsin Prognostic Breast Cancer (WPBC) data sets. Wine data sets consist of 13 inputs and three outputs, while WPBC data sets contain 33 inputs and two outputs.

Kyewords

Fuzzy Classification System, Hyper-cone Membership Function, Genetic Algorithms

1 Introduction

Fuzzy systems using fuzzy reasoning have been applied in various fields. However, there are tuning problems in membership functions and reasoning rules. Also, it is difficult to obtain fine fuzzy rules showing best performance for the system. Therefore, the study for automation of this processes have led to many researches with various tools for system development. For example, there are neural networks [1, 2, 3], genetic algorithms (GAs) [4, 5], clusterings [6, 7] and so on. Also, many methods of automatic generation for pattern classification problems using fuzzy systems have been proposed. Ishibuchi et al. were proposed GA based methods using various approach [8, 9, 10]. These method obtained compact fuzzy systems.

The purpose of this study is to obtain compact fuzzy classification systems dealing with high dimensional classification problems. The classification problem is a problem of estimating dimensional space and dividing the space into the regions of categories or classes. As the dimension increases, the system becomes complex.

In this paper, regarding to the classification problem, we present a method that combines fuzzy classification systems using hyper-cone membership functions [11, 12, 13] with GAs as the searching method. The hyper-cone membership function is expressed by a kind of radial basis function, and its fuzzy rule can be flexibly located in input and output spaces. Inoue et al. have proposed GA based automatic generation techniques for fuzzy rules using hyper-cone membership functions, and applied to vehicle navigations and inverted pendulum control problems [12, 14, 15].

We propose automatic generation methods of fuzzy classification rules using hyper-cone membership functions by GA. Dealing with high dimensional problems brings to a pre-assumption that some of the input vectors play important role in generating the rules, while others is assumed as unnecessary information. Therefore, the reduction of input number is considered to be important in this method. Also, unnecessary rules are eliminated to generate a compact fuzzy system. Therefore, to generate fuzzy classification systems, we propose an approach that refers to rule addition method. In this method, rule is added one by one, until the system reaches its best performance. The proposed method is applied to Wine Recognition Data and Wisconsin Prognostic Breast Cancer (WPBC) Data [16].

2 Fuzzy Pattern Classification Using Hyper-Cone Membership Functions

In [11, 12], fuzzy control rules using hyper-cone membership functions have been applied. In this paper, we propose fuzzy classification rules using hyper-cone membership functions. Therefore, fuzzy subsets A_i in the antecedent part is expressed by the hyper-cone membership function. Also, concept of "don't care input" is introduced, in order to reduce the number of inputs in each rule.

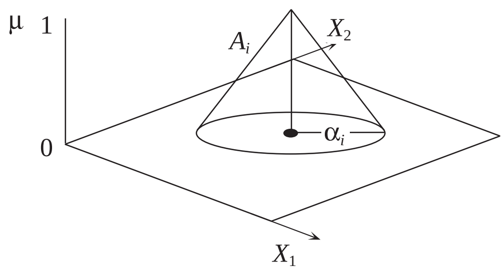

Figure 1: Shape of hyper-cone membership function ($l^i = 2$)

2.1 Fuzzy Classification Rules Using Hyper-Cone Membership Functions

In this method, the fuzzy rule R^i are expressed as follows:

$$R^i : \text{if } \boldsymbol{x}_i \text{ is } A_i \text{ then class is } C_i,\ i = 1, 2, \cdots, n \quad (1)$$

where i is rule number, n is the number of rules, $\boldsymbol{x}_i$ is the input vector of R^i, A_i is the fuzzy subsets of the antecedent part, and C_i is the output class. In this fuzzy system, the number of inputs (attributes) is L, and the input vector $\boldsymbol{x}$ for the fuzzy system is defined by Eq.(2).

$$\boldsymbol{x} = \left[\ x_1 \quad x_2 \quad \cdots \quad x_L\ \right]^{\mathrm{T}} \quad (2)$$

However, it seems that there exist important rules and unnecessary rules in input vector $\boldsymbol{x}$. Therefore, fuzzy rule use only necessary attributes, and other attributes are assumed “don’t care input”. Therefore, the input vector of each rule $\boldsymbol{x}_i$ is defined by Eq.(3).

$$\boldsymbol{x}_i = \left[\ x_{i1} \quad x_{i2} \quad \cdots \quad x_{il^i}\ \right]^{\mathrm{T}} \quad (3)$$

where l^i is the number of inputs of rule R^i.

The fuzzy subsets A_i of the rule R^i is expressed by hyper-spherical fuzzy subsets directly corresponding to a subspace in input space. Therefore, fuzzy subsets A_i is defined by a hyper-cone membership function described below. In this fuzzy system, since there are n fuzzy rules, n hyper cone membership functions are located in input space.

The hyper-cone membership function $\mu_{A_i}(\boldsymbol{x}_i)$ in l^i dimensional input space is defined by Eqs.(4) and (5).

$$\mu_{A_i} : A_i \rightarrow [0, 1] \quad (4)$$

$$\mu_{A_i}(\boldsymbol{x}_i) = \left(1 - \frac{\|\boldsymbol{x}_i - \boldsymbol{a}_i\|}{\alpha_i}\right) \vee 0 \quad (5)$$

where $\boldsymbol{a}_i$ and α_i are the center coordinate vector and the radius of the fuzzy subsets A_i, respectively. In Eq.(5), the center coordinate $\boldsymbol{a}_i$ are given by:

$$\boldsymbol{a}_i = \left[\ a_{i1} \quad a_{i2} \quad \cdots \quad a_{il^i}\ \right]^{\mathrm{T}} \quad (6)$$

The membership function μ_{A_i} has a grade 1.0 at the center $\boldsymbol{a}_i \in \boldsymbol{R}^{l^i}$ of the fuzzy subset A_i whose radius is α_i. The membership value decreases in proportion to the distance from the center $\boldsymbol{a}_i$. At the circumference of this sphere, a grade has 0.0. Figure 1 shows the hyper-cone membership function in case of $l^i = 2$.

The use of hyper-cone membership functions does not give the limitation to the size of membership functions. Therefore, hyper-cone membership function can be flexibly used even in the high dimensional cases.

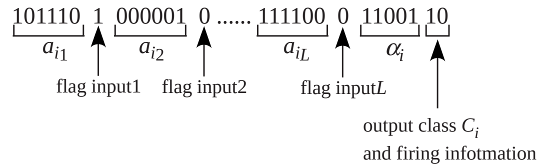

Figure 2: Chromosome of fuzzy rule R^i

2.2 Reasoning Method

Membership value μ_{A_i} of rule R^i for input vector $\boldsymbol{x}$ is determined by Eq.(5). Therefore, the truth value ω_i of each fuzzy rule is calculated as follows:

$$\omega_i = \mu_{A_i},\ i = 1, 2, \cdots, n \quad (7)$$

Output class C_{out} is class C_i having the maximum truth value ω_{out} (see Eq.(8)).

$$\omega_{out} = \bigvee_{i=1}^{n} \omega_i \quad (8)$$

3 Fuzzy Classification System Generation Methods by GAs

We generate fuzzy classification rules with hyper-cone membership functions by GAs. In this GAs process, we apply Pits approach method. In this method, one fuzzy system (a population of fuzzy rules) is made by one individual in the GAs process.

In this method, in order to obtain compact fuzzy systems for high dimensional problems, we propose a fuzzy rule generation method with the rule addition method.

3.1 Fuzzy Rule Generation Method Using GAs

3.1.1 Coding of Genes

Genetic information of fuzzy rule R^i is the location and the shape of input membership function and the output class. Therefore, parameters of hyper-cone membership functions composed one fuzzy rule are searched by GA. Genetic parameters of fuzzy rule R^i are as follow

- Center coordinate $\boldsymbol{a}_i$ of fuzzy subset A_i,
- Each input flag that is “don’t care input” or not,
- Radius α_i of fuzzy subset A_i,
- Output class C_i of fuzzy rule R^i, and
- Firing information.

In this information, each input flag is determined whether each input is used or not in a rule. If the flag of input j in rule R^i contains information that is considered as ‘un-used’ information, input data x_j is not used to calculate the membership value of rule R^i, in other words, the input is “don’t care input”. Also, firing information indicates for a sign whether rule R^i is fired or not in the fuzzy system. If this information is determined to be ‘not fired’, the rule R^i is not used in fuzzy reasoning. The genes of one chromosome for rule R^i is shown in Figure 2.

One individual I_p $(p = 1, 2, \cdots, P)$ consists of n fuzzy rules. So, one individual expresses n chromosomes in one strings. The population is composed of P individuals (see Figure 3). Then, P is the population size.

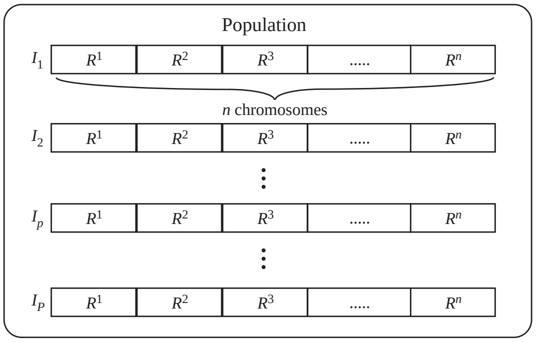

Figure 3: Composition of one population P

3.1.2 Fitness Function

In order to maximize the number of correctly classified patterns and minimize the number of fuzzy rules and inputs, the fitness fit_p for each individual is applied as Eq.(9).

$$fit_p = w_1 co_p - w_2 rn_p - w_3 in_p \tag{9}$$

where co_p is the number of correct patterns, rn_p is the number of the rules, and in_p is the number of used inputs. Then, w_1, w_2 and w_3 represent value of weight, respectively.

3.1.3 Genetic operation

The procedure of genetic operations is as follows:

Step 1: Initial population is generated. Generally, this process is carried out randomly. In the method described 3.2, however, some genes are defined some determined information. Also, some genes are fixed during GA operations.

Step 2: Evaluate the fitness of each individual I_p in the population.

Step 3: Create a new population by repeating following steps until a population size P is complete.

- **Selection:** Select two parent chromosomes from a population according to their fitness. This method applies the roulette wheel model.
- **Crossover:** A single point crossover method is conducted to form a new offspring.
- **Mutation:** With a mutation rate, mutate new offspring at each locus (position in chromosome).
- **Accepting:** Place new offspring in a new population.

Step 4: Use new generated population for a further run of algorithm when the number of population reaches P

Step 5: . If the process reaches G_{end} generation, the process stops. Otherwise return to Step 2.

3.2 Rule Addition Method

In this section, a rule addition method, that the fuzzy rules generation method described before section is effectively combined, is proposed. The proposed method refers to rule addition method, where rule is added one by one, until the system reaches its best performance. This method divides the process of GAs into two stages of process. Though the proposed method is quite simple, the process of GAs presents a solving for two-objective optimization problems: decrease the number of inputs and rules, and increasing the correct pattern number.

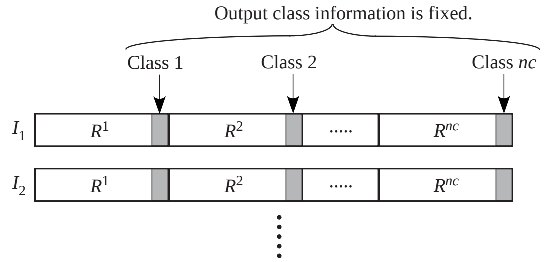

Figure 4: Composition of population in first stage

3.2.1 First Stage

In the first stage, minimum fuzzy rules are generated by GAs. In this stage, one fuzzy rule is generated in each class. Therefore, the number of fuzzy rules is the number of classes, and only main rule of each class is obtained by GA process.

If the number of classes is nc, the number of rules n in this stage is nc. Then, fuzzy rule R^i is the rule to determine output class i. So, it needs not to decide output class by GA, because output class of each rule is fixed. Therefore, only the location and shape of hypercone membership function in the antecedent part are assigned by GAs.

The genes of one individual for rule R^i in the first stage is shown in Figure 4. In this figure, some genes expressed output class of each rule are fixed. These genes are provided class information directly, and other genes are generated randomly. This gene strings is one individual I_p, and population is composed of P individuals. Though genetic operations described in 3.1 are implemented using this population, the mutation operation is not carried out for output class genes, because output class of each rule is fixed. If generation number reaches G_{1st}, genes string having maximum fitness value during all generations is defined as maximum gene string S_{max}, and continue to the second stage.

In this stage, fuzzy systems, which are one rule par one class, and comparatively high correct pattern number, are obtained.

3.2.2 Second Stage

In this stage, each one fuzzy rule is added to fuzzy rules composed of the maximum gene string S_{max}, and GA operation is executed. Therefore, genes of one fuzzy rule are randomly generated, and combined the maximum gene string S_{max}.

The composition of individuals in population is shown in Figure 5. In this stage, chromosome of one individual consists that genes of S_{max} are used the first n rules as it is, and genes of the $n+1$th rule are generated randomly. This gene strings is one individual I_p, population is composed of P individuals, and genetic operations described in 3.1 are implemented using this population. In initial population, the first n rules of all individuals are same because of using data of S_{max}. Also, genetic operations

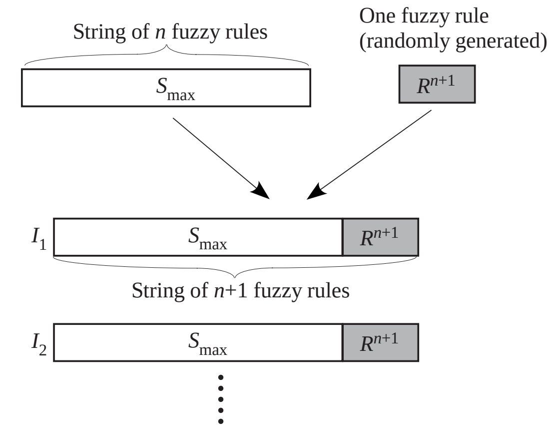

Figure 5: Composition of individuals in second stage

are carried out for all genes of each rule, though the mutation operation is not carried out for output class genes in the first stage. So, the location and shape of hyper-cone membership function in the antecedent part and the class in the consequent part of that rule are assigned by GAs.

After G_{2nd} generations, genes string having maximum fitness value during G_{2nd} generations is defined as maximum genes string in the second stage S_{max}^{2nd}. If S_{max}^{2nd} shows better performance than S_{max}, the maximum gene string S_{max} is exchanged with S_{max}^{2nd}.

This procedure are iterated to T_{2nd} times. Finally, the fuzzy rules of S_{max} is the obtained fuzzy system by GAs.

3.2.3 Procedure of Rule Addition Method

The procedure of the rule addition method with the first stage and the second stage is following:

[**First Stage**]

Step 1: Generated initial population like Figure 4.

Step 2: Genetic operations are curried out.

Step 3: After G_{1st} generations, genes string having maximum fitness value during G_{1st} generations is defined as maximum gene string S_{max}.

[**Second Stage**]

Step 4: Initial population like Figure 5 is generated using genes of S_{max}.

Step 5: Genetic operations are curried out.

Step 6: After G_{2nd} generations, genes string having maximum fitness value during G_{2nd} generations is defined as maximum gene string S_{max}^{2nd}.

Step 7: If S_{max}^{2nd} shows better performance than S_{max}, the maximum gene string S_{max} is exchanged with S_{max}^{2nd}.

Step 8: If iteration number reaches T_{2nd}, then go to Step 9. Otherwise, go to Step 4.

Step 9: the fuzzy rules of S_{max} is the obtained fuzzy system by GAs.

Table 1: Setting of parameters (Wine data)

	Proposed Method	Comparative Method
Generation parameter	G_{1st} 5000 G_{2nd} 1000 T_{2nd} 5	G_{end} 10000
Population size P	30	30
Initial rule number	3	30
Weight w_1	1.23×10^{-2}	1.38×10^{-2}
Weight w_2	0.0	4.17×10^{-2}
Weight w_3	2.85×10^{-3}	3.21×10^{-3}

By applying this method, it does not require such a long process time as the system just focuses on the searching of two parameters: number of correct pattern and number of input. In this method, the rule number has already been fixed in the first stage. For that reason, number of genetic information that is used for one chromosome is much less. In the first stage, one chromosome is built based on the minimum rules. Therefore the process time significantly reduces as the searching just need to find the best combination between a few numbers of parameters and numbers of bits.

4 Application to Pattern Classification Problems

The proposed method was applied to Wine Recognition Data and Wisconsin Prognostic Breast Cancer (WPBC) Data. These data are the well-known data set achieve that is taken from UCI Repository of Machine Learning Database [16].

For the comparison, the fuzzy rule generation method without the rule addition was applied to these data (comparative method). So, this is used only the GA based method described 3.1. Therefore, in initial GAs setting, many rules are randomly generated, and the number of rules is decreased by effectiveness of fitness function.

4.1 Application to Wine Data

Wine Data contains 178 data sets with 13 attributes (inputs) and three classes. In these data sets, we specify 50% of data sets on each class as training data, and the rest 50% of data set on each class as testing data. Consequently, 90 data sets were used as training data and the rest 88 data sets as testing data.

Simulation parameters of both methods are shown in Table 1. Also, the mutation rate was set into 0.01. Using these parameters, ten simulations for each method were run to evaluate each system.

Table 2 shows simulation results of Wine Data. In proposed method, the average number of rules is 3.7. Also, the average number of inputs is 2.6. By using the input flag, the rest 10.40 inputs are considered as "don't care inputs". These values are same or smaller than results of the comparative method. Consequently, this method generates compact fuzzy systems consist of rules with few inputs. Also, the simulation shows that the average rate of correct pattern classification for training data reaches 98.60%, and it reaches 97.70% for the testing data. Therefore, proposed method is shown better performance than the comparative method.

Table 2: Simulation result of Wine data

		Proposed Method	Comparative Method
Training data (90)	Average	88.7(98.60%)	84.8(94.22%)
	Maximum	90(100.00%)	86(95.56%)
Test data (88)	Average	81.8(92.95%)	78.3(89.00%)
	Maximum	86(97.70%)	84(95.45%)
The average number of rules		3.7	3.5
The average number of inputs of each rule		2.6	3.3

Table 3: Setting of parameters (Breast cancer data)

	Proposed Method	Comparative Method
Generation parameter	G_{1st} 3000 G_{2nd} 1000 T_{2nd} 2	G_{end} 5000
Population size P	30	30
Initial rule number	2	30
Weight w_1	1.11×10^{-2}	1.20×10^{-2}
Weight w_2	0.0	3.88×10^{-2}
Weight w_3	1.09×10^{-3}	1.17×10^{-3}

In Wine Case, as the consequent part comprises of three classes, the minimum value of rule number is three. Most of the simulations reach their best performance on the first stage, while the second stage plays important in performance improvement. Therefore, the system is avoided from further rule increment. The numbers of inputs significantly decrease. The proposed method obtains the fuzzy systems with small number of inputs in each rule.

4.2 Application to Breast Cancer Data

WPBC Data contains 198 data sets, 33 inputs and two classes. However, since four data sets are assumed to be missing, we eliminated those data sets from the simulation process. In these data sets, we specify 50% of data sets on each class as training data, and the rest 50% of data set on each class as testing data. Both training data and testing data on WPBC Case is set into 97 data sets.

Simulation parameters are shown in Table 3. Also, the mutation rate was set into 0.01. Using these parameters, ten simulations for each method were run to evaluate each system.

Table 4 summarizes the simulation result of WPBC Data. As most of the best result is gained at the first stage where the number of rules in the first stage is set to be two, the simulations reached their best performance at the average of 2.00 rules. By using the input flag in the GA process, about 7.48 inputs is used in a rule. In these results, this method generates more compact fuzzy systems of WPBC data classification than the comparative method.

The simulation result shows that the average rate of correct pattern classification for training data reaches 76.29%, and it reaches 76.19% for the testing data. Therefore, proposed method is shown better performance than the comparative method a little.

As the consequent part comprises of two classes, the minimum value of rule number is two. The numbers of rules can be reduced into the minimum number. The best performance of the system is found in the first stage (where rule is fixed by two).

However, about 24% of data sets fails to be covered by the rules. It seems that the minimum fuzzy systems with a certain level of accuracy are obtained by the first stage GA operations. However, in the second stage, additional rules to improve the system performance are not found. It is necessary to consider methods for adding effective rules in the second stage.

5 Conclusion

In this paper, we presented a new approach regarding to the high-dimensional pattern classification problems. We proposed an automatic generation method of fuzzy classification system using hyper-cone membership functions by GAs. Also, the proposed method referred to rule addition method. In this method, rule was added gradually until the best performance was achieved. This method is applied to classification problems of Wine Data and WPBC Data. In simulation results of Wine Data, effective and compact fuzzy systems were obtained. Also, in the WPBC Data, simulation results of presented method were shown better performance than the comparative method. However, it is necessary to consider methods for adding effective rules in the second stage.

In the future work, we would like to improve the feasibility of the fuzzy systems using the GAs to simulate some parameters. Also, the combination of hyper-cone membership function with other searching method, as Neural Network, or combination both GAs and Neural Network is considered. The purpose is to find the best combination among number of the correct pattern, number of rule and number of input to perform maximum result.

References

[1] S. Horikawa, T. Furuhashi, and Y. Uchikawa. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm. *IEEE Transactions of Neural Networks*, 3(5):801–806, 1992.

[2] I. Hayashi, H. Nomura, H. Yamasaki, and N. Wakami. International journal of approximate reasoning. *Construction of Fuzzy Inference rules by NDF and NDFL*, 6(2):241–266, 1992.

[3] J. A. Dickerson and B. Kosko. Fuzzy function approximation with ellipsoid rules. *IEEE Transactions of Systems, Man, Cybernetics–B*, 26(4):542–560, 1996.

[4] M. A. Lee and H. Takagi. Integrating design stage of fuzzy systems using genetic algorithms. In *Proceedings of 2nd IEEE International Conference on on Fuzzy Systems*, pages 612–617, 1993.

Table 4: Simulation result of WPBC data

		Proposed Method	Comparative Method
Training data (97)	Average	74.0(76.39%)	73.5(75.77%)
	Maximum	75(77.32%)	74(76.29%)
Test data (97)	Average	73.9(76.19%)	71.0(73.20%)
	Maximum	74(76.29%)	73(75.26%)
The average number of rules		2.0	3.7
The average number of inputs of each rule		7.48	12.20

[5] K. Shimojima, T. Fukuda, and Y. Hasegawa. Rbf-fuzzy system with ga based unsupervised/supervised learning method. In *Proceedings of 4th IEEE International Conference on on Fuzzy Systems*, pages 253–258, 1995.

[6] K. Kamei. An application of fuzzy clustering to controller design. *Journal of Japan Society for Fuzzy Theory and Systems*, 8(3):448–455, June 1996 (in Japanese).

[7] M. Ryoke, H. Tamura, and Y. Nakamori. Fuzzy rule generation by hyperellipsoidal clustering. In *Methodologies for the Conception, Design and Application of Intelligent Systems (World Scientific)*, pages 86–89, Iizuka, Japan, October 1996.

[8] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka. Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. *Fuzzy Sets and Systems*, 65(2), 1994.

[9] T.Murata, H.Ishibuchi, and M.Gen. Adjusting fuzzy partitions by genetic algorithms and histograms for pattern classification problems. In *Proceeding of 1998 IEEE International Conference on Evolutionary Computation*, pages 9–14, Anchorage, USA, May 1998.

[10] T.Nakashima, H.Ishibuchi, and T.Murata. Evolutionary algorithms for constructing linguistic rule-based systems for high-dimensional pattern classification problems. In *Proceeding of 1998 IEEE International Conference on Evolutionary Computation*, pages 752–757, Anchorage, USA, May 1998.

[11] H. Inoue, K. Kamei, and K. Inoue. Auto-generation of fuzzy production rules using hyper-cone membership function by genetic algorithm. In *Fuzzy Logic for the Applications to Complex Systems (World Scientific)*, pages 53–58, Taipei, December 1995.

[12] H. Inoue, K. Kamei, and K. Inoue. Automatic generation of fuzzy if-then rules using genetic algorithms and hyper-cone membership functions. *Japanese Journal of Fuzzy Theory and Systems*, 8(6):957–975, December 1996.

[13] H. Inoue, K. Kamei, and K. Inoue. Automatic generation of fuzzy rules using hyper elliptic cone membership functions by genetic algorithms. *Journal of Intelligent and Fuzzy Systems*, 6(1):65–81, July 1998.

[14] H. Inoue, K. Hatase, and K. Kamei. Fuzzy classifier system using hyper-cone membership functions and rule reduction techniques. In *Proceedings of The 10th IEEE International Conference on Fuzzy Systems*, pages 1436–1439, Melbourne, December 2001.

[15] H. Inoue, K.Matsuo, K.Hatase, K.Kamei, M.Tsukamoto, and K.Miyasaka. A fuzzy classifier system using hyper-cone membership functions and its application to inverted pendulum control. In *Proceeding of 2002 IEEE Conference on Systems, MAN and Cybernetics (CD ROM)*, page Paper Number WA2D3, Hammamet, Tunisia, October 2002.

[16] [UCI Repository of Machine Learning Databases Network Document]. Available: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/.