International Journal of Fuzzy Logic and Intelligent Systems, vol. 2, no. 2, June 2002 pp. 159-165

# Selecting Fuzzy Rules for Pattern Classification Systems

**Sang-Bum Lee, Sung-joo Lee, Mal-Rey Lee**

**Department of Computer Science, Chosun University**

Department of Multimedia Information & System, Yosu Natinonal University

**Abstract**

This paper proposes a GA and Gradient Descent Method-based method for choosing an appropriate set of fuzzy rules for classification problems. The aim of the proposed method is to find a minimum set of fuzzy rules that can correctly classify all training patterns. The number of inference rules and the shapes of the membership functions in the antecedent part of the fuzzy rules are determined by the genetic algorithms. The real numbers in the consequent parts of the fuzzy rules are obtained through the use of the descent method. A fitness function is used to maximize the number of correctly classified patterns, and to minimize the number of fuzzy rules. A solution obtained by the genetic algorithm is a set of fuzzy rules, and its fitness is determined by the two objectives, in a combinatorial optimization problem. In order to demonstrate the effectiveness of the proposed method, computer simulation results are shown.

**Key words** : genetic algorithm, gradient descent method, classification problems.

## 1. Introduction

The study of information processing systems based on fuzzy rules has, thus far been mainly applied to control problems.1, 2 The fuzzy rules that are used in most fuzzy control systems are generally derived from human experts experience, using linguistic information. A high non-linear system has under gone many trials and errors and several experiments to acquire proper fuzzy rules. It provided fuzzy rules using numerical information of I/O data, doing the study to use a learning, and having been the study used the neural network and clustering.3, 4 For classification problems, the automated generation method of fuzzy rules has been proposed by Ishibuchi et al.5, 6 The generation of fuzzy rules from numerical data for pattern in classification problems consists of two phases: fuzzy partition of a pattern space into fuzzy subspaces, and the determination of a fuzzy rule for each fuzzy subspace. The pattern spaces are divided by a fuzzy grid, and the fuzzy rules are generated in each the fuzzy spaces. The performance of a fuzzy classification system based on fuzzy rules depends on the choice of the fuzzy partitions. If the partition is too coarse, the performance may be low (many patterns may be unclassified). On the other hand, if the fuzzy partition is too fine, many of the fuzzy rules needed may not be generated, because of the lake of training patterns in the corresponding fuzzy subspaces. Therefore the choice of the fuzzy partition 4a very important stages one. For example, consider the two-class classification problem shown in Fig. 2., where closed circles and open circles denote the patterns in class 1 and class 2, respectively.

Because the choice of an appropriate fuzzy partitioning based on a simple fuzzy grid is make more complicated by the difficulty of finding simple fuzzy grid, the concept of distributed fuzzy rules was proposed in 4, 5, where all the fuzzy rules corresponding to several fuzzy partitions were simultaneously employed in a fuzzy classification system. The main drawback to this approach is that the number of fuzzy rules is enormous. If unnecessary fuzzy rules are removed, and only relevant fuzzy rules are selected, however the performance of the selected rule set may be high with far fewer fuzzy rules.

This paper proposes a GA and GDM-based method for removing the unnecessary rules and generating the relevant rules from the fuzzy rules corresponding to several fuzzy partitions. The aim of the proposed method is to find a minimum set of fuzzy rules, that can correctly classify all the training patterns. This is achieved by formulating and solving a combinatorial optimization.problem that has two objectives: to maximize the number of correctly classified patterns, and to minimize the number of fuzzy rules.

A fine fuzzy division needs to be chosen, in consist of good fuzzy value classification system. As a solution to this problem, a concept of distributed fuzzy rules using a large number of fuzzy divisions simultaneously, and a fuzzy classification system can be employed.6 This method solves the problem of generating a fine fuzzy division but degrades the system efficiency on account of using many of fuzzy rules in inference.

In the proposed method, the fuzzy rules corresponding to various fuzzy partitions are simultaneously utilized in the fuzzy inference process.7

Fuzzy rules are generated from an area with the highest inference errors among those areas which are divided by two membership functions of neighboring antecedent part. The fuzzy of inferences are structured by a set of simple fuzzy rules. In each rule, the antecedent part is made up the

Manuscript received February 2002 ;
revised May 2002

membership functions of a fuzzy set, and the consequent part is made up of a real number. The membership functions and number of fuzzy inference rules are tuned by means of the Gas, while the real numbers in the consequent parts of the rules are tuned by means of the gradient descent method.11 GAs solve problems by using principles inspired by natural selection: they maintain a population of knowledge structures that represent candidate solutions, and then let that population evolve over time through competition and controlled variation.8, 9, 10

## 2. Simplified fuzzy reasoning

This section explains the method used to control the real numbers in the consequent parts of the rules by using simple inference and the gradient descent method.

When the inputs are expressed as $x1$, $x2$, ·, xm, and the output is expressed as y, an inference rule used in simplified fuzzy reasoning can be expressed as the follows. 12

Rule $i$ : IF x1 is Ai1 · and xm is Aim THEN y is Awi

$$(i=1,\cdots,n) \tag{1}$$

where i is the rule number, Ai1, ·, Aim are the membership functions of the antecedent part, and wi is a real number in the consequent part.

The membership function Aij($xj$)of the antecedent part is represented by an isosceles triangle as shown in Fig. 1.

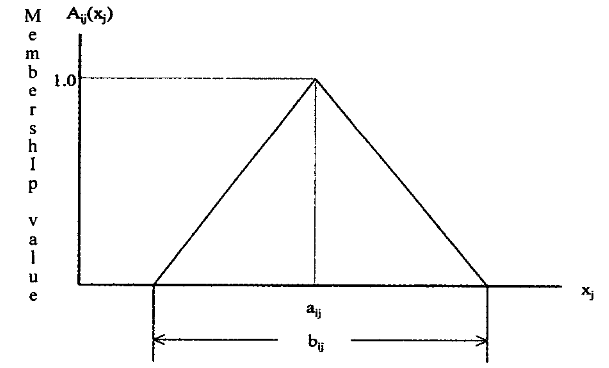



Fig. 1. Membership function of antecedent part

The parameters determining the triangle are the center value aij and the width bij. The output of the fuzzy reasoning, y, can be derived from the equations shown below.

$$A_{ij}(x_j)=1-\frac{2\cdot|x_j-a_{ij}|}{b_{ij}}\quad(j=1,\cdots m)\in[0,\ 1] \tag{1}$$

$$\mu i=Ai1(x1),\ Ai2(x2)\cdots Aim(xm) \tag{3}$$

$$y=\frac{\sum_{i=1}^{n}\mu i\cdot wi}{\sum_{i=1}^{n}\mu i} \tag{4}$$

where $\mu$i is the membership value of the antecedent part. This was used as an entity (member of the population) in the genetic algorithms. A real number in the consequent part, wi, is optimized by using the gradient descent method to provide a local fine tuning mechanism for the GA.

The gradient descent method is designed to seek for the vector Z that minimizes an objective function $E(Z)$, where Z is a p-dimensional vector $Z=(Z1,\ Z2,\cdots ZP)$ of the tuning parameters.11 In this method, the vector that decreases the value of an objective function $E(Z)$ is expressed as $(-\partial E/\partial Z1,\ -\partial E/\partial Z2,\cdots,\ -\partial E/\partial Zp)$, and the learning rule is expressed by the following formula.

$$Zi(t+1)=Zi(t)K\frac{\partial E(Z)}{\partial Z_i}(i=i,\cdots,p) \tag{5}$$

where t is a number of iterations of learning, and k is a constant. By altering Z according to this learning rule, the value of the objective function E(Z) converges to a local minimum.13

In the present method, the inference rules are tuned so as to minimize the objective function E which is defined as follows.

$$E=\frac{1}{2}(y-y^p)^2 \tag{6}$$

where yp is the desirable output data (as acquired from specialists). The objective function E represents the inference error between the desirable output yp, and the output of fuzzy reasoning, y.

The objective function E consists of the tuning parameter wi. From Eq. 5, the learning rules of simplified fuzzy reasoning are expressed by Eq.7.

$$w_i(t+1)=w_i-k_w\frac{\partial E}{\partial w_i} \tag{7}$$

Eq.7 show respective (t+1)th values of tuning parameter kw is constant. The learning rules Eq.7 is to adaptively change the tuning parameters for a direction to minimize the objective function E. Thus, using the learning rules of Eq.7, the tuning parameter of inference rules is optimized to minimize the inference error between the desirable output yp and the output of fuzzy reasoning y.

When repeatedly applying I/O data to the fuzzy rule, we can minimize the object function and acquire a global fitness solution without falling in to the local minimum value.

Conventional self-tuning methods need many experiments, trials and error process in order to search for optimal rules. This paper uses a GA and the GDM to acquire the optimal rules.

## 3. Optimization of fuzzy rules

This section explains a method of optimizing the membership function shape and the number of fuzzy inference rules using GAs. The real numbers of the consequent parts of the fuzzy rules are obtained through the use of the gradient descent method. GAs are an optimization technique based loosely on the principles of natural selection. GAs start with a set of encoded parameter strings, and an evaluation of the parameter performance corresponding to each string. Then, through the operations of reproduction, crossover and mutation, the algorithm attempts to introduce increasingly fit strings into the set. Mutation is added after crossover to expand the region of points that can possibly enter as members of the population. The GAs choose the string with the maximum fitness function E(sr). Each string is represented as a binary number. A set of string S, called the population, can be represented as

$$sr = L\,r1,\ L\,r2,\ Lrg(g=1,\cdots,G) \quad (8)$$

$$S = s1,\ s2,\cdots,sR \quad (9)$$

### 3.1 Generation of fuzzy rules

This paper consider the second group in the classification problem shown in Fig. 2. The figure shows examples of generated problem instances for the after eq. $f(x)=1/4\sin(2\pi x1)+x2-0.5$. The pattern space [0,1] * [0,1] is divided into two classes according to the value of the following function $f(x) = 1/4\sin(2\pi x1)+x2-0.5$. If a pattern has $f(x)$ 0, then $x$ belongs to $G1$, otherwise x belongs to $G2$, Closed circles and open circles in Fig.2 represent patterns belong to $G1$ and $G2$ respectively. As learning data, we divided the group numbers of M $(G1, G2,\cdots, GM)$ and supposed the pattern number of M( $xp=(xp1, xp2,\cdots, xpm)$, $p=1,2,\cdots,m)$. Then each dimension of 2-dimensional pattern space consists of a fuzzy set of the number of k.

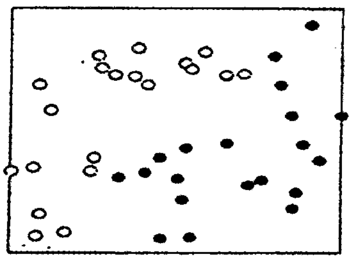


Fig. 2. A classification problem

**[Generation method for fuzzy rules]**

**[Step 1]** Determination of generation of regions for the fuzzy rules.

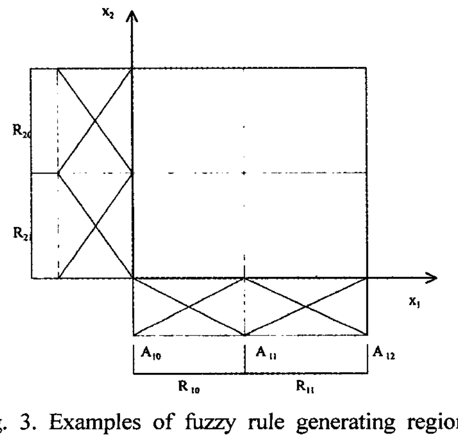



Fig. 3. Examples of fuzzy rule generating regions

A rule is generated from the regions with the biggest inference error, which is calculated in an each area partitioned by membership functions in neighboring two-antecedent part. Fig.3 shows an example for rule generation regions. The three-membership function in Fig. 3 is a fuzzy set with the input variables x1 and $x2$. An inference error has four areas; that is to say, it is calculated from R10, R11 for $x1$, R20 and R21 for $x2$. The region for rule generation is also selected for each input variable.

**[Step 2]** Membership function generation of the antecedent part

Membership functions are separately generated into two areas of a central value by step 1. Fig. 4 shows as an example of membership functions generate by area R10.

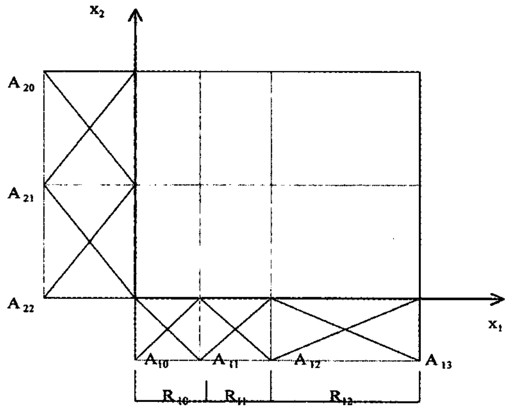



Fig. 4. Example of the generated Rules

**[Step 3]** Self-tuning a real number of the consequent part

Self-tuning is achieved by means of the gradient descent method of eq. 7. The membership functions of an antecedent part are represented in Fig. 5.

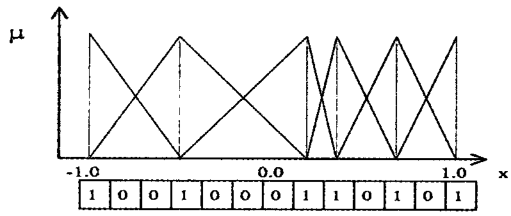



Fig.5. String representation of membership functions

The binary representations of the membership function consist of strings of 1, and 0, . A center value appears as 1, and other appear as 0, . The optimal membership function numbers and its center value toward to each input variable xj searched by the GA. The fitness $E(sr)$ which is able to maximize the center value and the number of membership functions are searched for using a GA as follows.

#### A. A fitness definition

This paper use the learning pattern number $C(sr)$ while representing the number of patterns that are classified by the rule set in a evaluation function employed for solving a classification problem using the GA. The number of rule sets represented $|sr|$, the fitness function is defined by the maximum $C(sr)$ and minimum $|sr|$. The fitness function $E(sr)$ can be derived from the equation shown below.

$$E(sr) = \max Wc * C(sr) Wsr * |sr| \quad (10)$$

**B. Definition of a entity**

In order to find the optimal solution by using the GA, the entities have to be expressed as strings in order to find an executable solution. In this paper, the number of rules and the membership function are represented by several long strings, which are treated as entities.

**C. Operations of the GA**

This paper uses a simple genetic algorithm employs a GA that extracts the rule by using an eliteness preservation strategy. The GA consists of five basic operations. The following operations are applied to a set of individuals (i.e., the population in a generation) in order to generate a new population in the next generation:

**←Creation of an initial population .**

The center values of the membership functions and the widths of the neighboring membership functions are initialized as $aij=1$ and $bij=0$ respectively. To evaluate the initial population, the function representing the environment is evaluated for the values encoded in each of the n strings in the initial population p(0).

**↑ Selection operation**

To create the next generation by a crossover operation, a selection probability Psr(t) is represented as follow.

$$P_{sr}(t) = \frac{E(_{sr}(t))}{\sum_{r=1}^{r} E(s_r(t))} \quad (11)$$

**→ Crossover operation**

The operation is described as follows: only two strings in a generation are selected. And two populations in the string, that constructs the base of a membership function, are also selected. The string values between these two positions are exchanged between the two string. The offspring generate four species by means of crossover point the operation. The way to generate offspring 1 is as follows: The values of the left side of the string are inherited from one string (parent 1) and the values on the other side are inherited from the other string (parent 2), where each according to crossover probability value. The new generation the group entity is iterative generated by this crossover operation .

**↓ Mutation operation**

The mutation operation in this paper operates in such a way that the selected membership function in pruned at probability Pm. One can expect to reduce the number of the fuzzy rules, and to obtain the maximum of effectiveness in solving classification problem by the mutation operation. The following mutation operation is applied to each bit of the individuals generated by the crossover operation: sr sr * (-1). Each bit of each individual undergo this mutation operation with the mutation probability Pm.

**Elite preservation strategy**

The best individual (i.e., the individual with the largest fitness value) in each generation always survives and exists in the next generation. This elite preservation strategy has the merit that superior strings do not undergo the GAs operations.

**3.2 Self-tuning procedure**

The procedure to acquire an optimal fuzzy rule using a GA is as follows.

Step 1 : Randomly generate all entities (population) $sr$(t), $r=1,\cdots$, R about a initial generation ($t=0$).

Step 2 : Decide the initial real number of the consequent part, using the gradient descent method.

Step 3 : Select two entities $si(t)$ and $sj(t)$ from population S(t) according to selection probabilities $Psi(t)$ and $Psj(t)$.

Step 4: Perform a crossover operation to create a new entity sk(t) from the two selected entities.

Step 5: Perform a mutation operation on a string of an entity sk(t), using the mutation probability Pm.

Step 6: Until the new number of entities k, becomes equal to R, repeat from step 3 to step 5.

Step 7: A new group (population) $S(t+1)=\{s1(t), s2(t), \cdot, sR(t)\}$ is generated from step 3 to step 6.

Step 8: Add 1 to the generation number t, and repeat from step 2 to step 8 until the population S converges. The largest fitness entity in the converged group becomes an optimal solution.

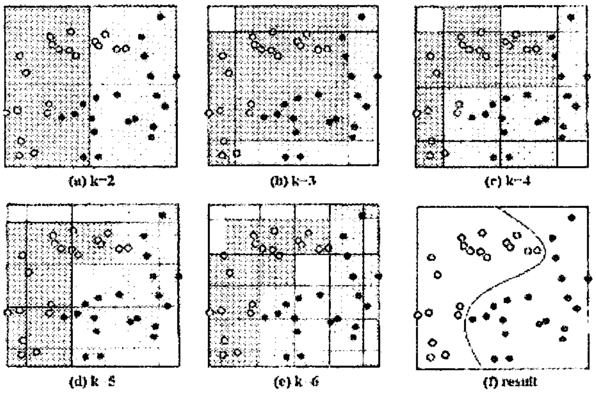
## 4. Results of computer simulations

In this section, the new method is compared with Ishibuchis method by means of an experiment.4

4.1 Rule creation and pattern classification by a fuzzy grid

A classification problem is applied to Fig. 2. Ishibuchi divided the fuzzy space into subspaces using a fuzzy grid and spaces generating fuzzy rules. Fig. 6 shows to be achieved fuzzy division to six-fuzzy-set each dimension of two dimension pattern spaces.

If the fuzzy division is small (k is larger) then many patterns may be classified, and many fuzzy rules can be generated, but the performance may be low. On the other hand, if the fuzzy division is large (k is smaller) then many fuzzy rules can not be generated because of the lack of training patterns in the corresponding fuzzy subspaces.



Black area: the rules of the first group (white dots) : $G1$
Gray area: the rules of the second group (black dots) : $G2$

Fig. 6. Generated fuzzy rules and classification results

### 4.2 Rule creation and pattern classification by GA

A genetic algorithm with the following parameter specifications was applied to the classification problem in Fig.2.

Population size in each generation($R$ ) : 20 individuals
Stopping Condition($t$) : 1000 generation
Mutation Probability($Pm$) : 0.01
Length of entity($G$) : 13
Value of critical : 1.0*10−5

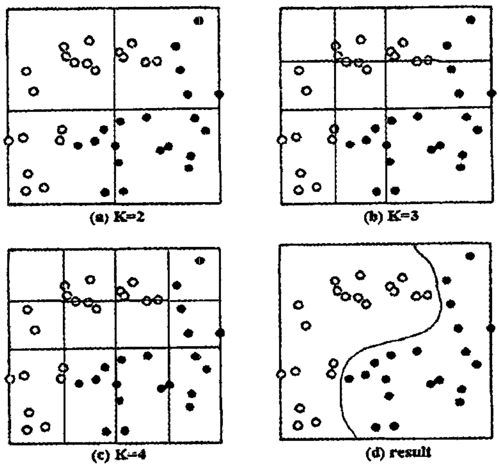


Fig. 7. Simulation results using the proposed method

From the classification results of Fig. 6, when k equals 6, it is possible to see that the whole pattern can be classified. In this case, the total number of rules 60. The executable numbers of the solution becomes =1.2*1017. If a GA is applied, then a total of 2000 rules are generated (20 individual *1000 generations). After these values were set, the first population was randomly changed 20 times. Fig. 7 shows the rules created and the classification results, using a numerical value experiment.

Comparing Fig. 6. with Fig. 7, it is possible to understand the pattern classifications when k equals 4 with even smaller than in Fig. 6. The areas (a)-(c) in Fig. 7 show the fuzzy rules generated by a GA, where (d) is the result of classifying the whole learning pattern. The method was there applied to some searched classification problems as shown in the next section.

### 4.3 Test problems

As test problems, five classification problems were selected. In each problem, the pattern space [0,1]*[0,1] is divided into two classes according to the value of the following function f(x), i.e., if $f(x)$ 0 then x belongs to $G1$, otherwise x belongs to $G2$.

Problem 1 : $f(x)=-1/4\sin(2\pi x1)+x2$ 0.5
Problem 2 : $f(x)=-1/3\sin(2\pi x1)+x2$ 0.5
Problem 3 : $f(x)=-1/3\sin(2\pi x1$ $1/2\pi)+x2$ 0.5
Problem 4 : $f(x)=-|-2x1+1|+x2$
Problem 5 : $f(x)=(x1+x2$ $1)(-x1+x2)$

For each classification problem, 20 problem instances were randomly generated, where each of 10 problem instances has 20 patterns in each class as the given patterns (i.e., as the training patterns). Closed circles and open circles in the given patterns belong to $G1$ and $G2$, respectively. These patterns are used for deriving the fuzzy rules in computer simulations. Table 1.2 shows the usefulness of the suggested method (e. g., in Problem 1).

Table 1. The number of classified patterns and fuzzy rules

| Conventional method | | |
|---|---|---|
| No of patterns | Generated Rules | Classified No |
| 20 | 46 | 20 |
| 40 | 60 | 40 |
| 100 | 92 | 100 |
| 200 | 117 | 200 |

Table 2. The number of classified patterns and fuzzy rules

| Suggested method | | |
|---|---|---|
| No of patterns | Generated Rules | Classified No |
| 20 | 12 | 20 |
| 40 | 25 | 40 |
| 100 | 37 | 100 |
| 200 | 58 | 200 |

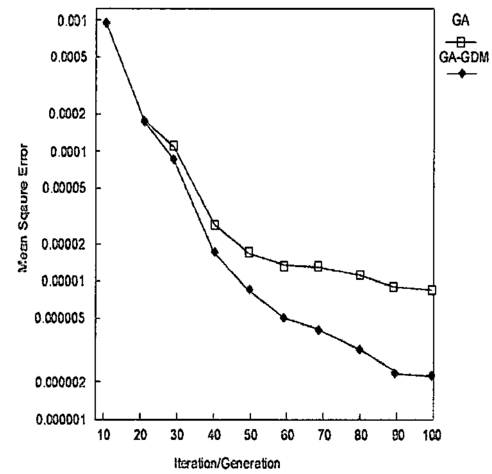



Fig. 8. Results after 100 epochs of learning

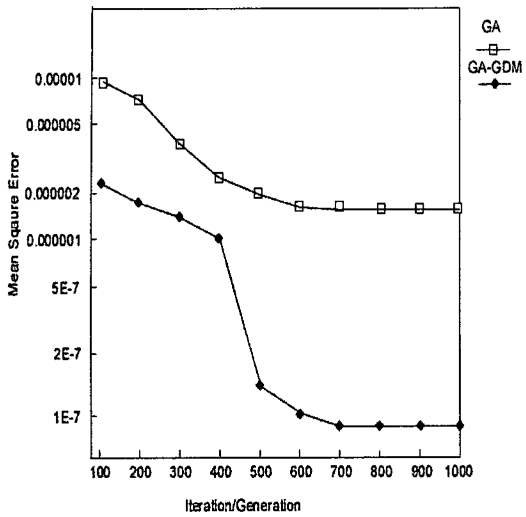



Fig. 9. Results after 1000 epochs of learning

In the application to the iris data,4 was the following biased mutation probability employed in order to reduce the number of fuzzy rules by the mutation operation: $Pm=0.01$ for the mutation from $Sr=1$ to $Sr=-1$, and $Pm=0.001$ for the mutation from $Sr=-1$ to $Sr=1$.

## 5. Conclusion

The GA and gradient descent method-based rule generation method has been proposed here for finding an appropriate set of fuzzy rules for classification problems. In the proposed method, a GA and gradient descent method were applied to the generation of fuzzy rules generated from numerical data. A combinatorial optimization problem was formulated for finding a minimum set of fuzzy rules that can correctly classify all the given patterns. The GA and gradient descent method were applied to this problem, and simulation results were shown. But we had only one among probability values of mutation and took a our examination. Future work will involve studying the effects of various probability values on classification problems.

## References

[1] K. H Lee, K. R Oh, (1991), Fuzzy Theory and Applications I-II, Hongrung Press.

[2] H. Ichihshi and T. Watanabe, (1990), Learning control system by a simplified fuzzy reasoning model , IPMU90, Paris-France, July 2-6, pp. 417-419.

[3] Shigeo and Ming-Shong Lan, (1995), A method for fuzzy rules extraction directly from numerical data and its application to pattern classification , IEEE Transactions on Fuzzy System, Vol. 3, No. 1, pp. 18-28.

[4] H. Ishibuchi, K. Nozaki and H. Tanaka, (1992), Efficient fuzzy partition of pattern space for classification problems. Proc. of the Second Interrational Conference on Fuzzy Logic & Neural Networks (Iizuka, JAPAN), pp. 671-674.

[5] H. IIIlshibuchi, K Nozaki and H. Tanaka, (1992), Distributed representation of fuzzy rules and its application to pattern classification, Fuzzy Sets and Systems, Vol. 52, pp. 21-32.

[6] H. Ishibuchi, K. Nozaki and R. Weber, (1992), Approximate pattern classification with fuzzy boundary , Proc. of International Joint Conference on Neural Networks, Vol. 52, pp. 21-32.

[7] D. E. Goldberg, (1989), Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts.

[8] Thrift P., (1991), Fuzzy logic synthesis with genetic algorithms , Proc. of the Fourth International Conference on Genetic Algorithm, San Diego, USA, pp. 509-513.

[9] C. L. Karr, (1991), Design of an adaptive fuzzy logic controller using a genetic algorithm, Proc. of the Fourth International Conference on Genetic Algorithms, pp. 450-457.

[10] Karr C., (1991), Genetic algorithms for fuzzy controllers , AI Expert, February, pp. 26-33.

[11] H. Nomura, I. Hayashi and N. Wakami, (1991), A self-tuning method of fuzzy control by descent method, Proc. of 4th IFSA Congress, Brussels, pp. 155-158.

---

**Sang-Bum Lee**
1988: Dept. of Computer Sciences, Kwangju University(B.S)
1994: Dept. of Computer sciences, Chosun University(M.S)
2001: Dept. of Computer sciences, Chosun University(ph.Co)

Research Interests :Software engineering,Computer Computer Communication,

**Sung-Joo Lee**
1970 : Dept. of Physics Sciences, Hannam University(B.S)
1992 : Dept. of Computer sciences, KwangwoonUniversity(M.S)
1998 : Dept. of Computer sciences, Catholic University of Daegu(Ph.D)

1988～1990 : Chief, Computer Center, Chosun University
1995～1997 : President, Information Science, Chosun University
1981～now : Professor in the Dept. of Computer Engineering, Chosun University
Research Interests : Software engineering,
Programming Language, Object-oriented software, Rough set.

**Mal Rey Lee**
1993-998, Ph.D computer engineering from Chung-Ang University, Seoul, Korea
1999-now Associate professor of the department of multimedia at Yosu National University in Korea.
Research Interests : Artificial intelligence, computer languages, multimedia science area. Currently