International Journal of Innovative Computing, Information and Control
Volume **9**, Number **4**, April **2013** ISSN 1349-4198
pp. 1593–1601

LEARNING WEIGHTS OF FUZZY RULES BY USING GRAVITATIONAL SEARCH ALGORITHM

Ersin Kaya*, Barış Koçer and Ahmet Arslan
Department of Computer Engineering
Faculty of Engineering and Architecture
Selçuk University
Konya 42075, Turkey
*Corresponding author: ersinkaya@selcuk.edu.tr

Received November 2011; revised March 2012

Abstract. *The fuzzy rules are the most important factor which affects the success in fuzzy rule-based systems. Performance of a fuzzy rule-based classifier can be improved by weighting fuzzy rules. There are different methods for weighting fuzzy rules. In this study, weights of fuzzy rules which are generated from datasets in fuzzy rule-based classifiers are determined by using Gravitational Search Algorithm. Gravitational search algorithm is a new algorithm for optimization problems. The aim of the GSA in the proposed method is to find rule weight values which maximize classification accuracy. The proposed method was tested with six different data sets. The simulation results are compared by the other method. The proposed method mostly provided better results than the other method compared.*
Keywords: Learning, Fuzzy rule-based classification, Gravitational search algorithm, Weighting fuzzy rules

1. **Introduction.** In fuzzy rule-based classification systems (FRBCSs), fuzzy if-then rules and fuzzy reasoning are used for classification pattern. FRBCSs have advantages on solution of the problems which include uncertainty. However, the need for generating fuzzy rule set by experts is a disadvantage of this system. Many approaches have been proposed for generating and learning fuzzy if-then rules from numerical data for classification problems like evolutionary algorithms [1-4], neuro-fuzzy hybrid approaches [5-7] and data mining techniques [8,9].

In FRBCSs, system performance has been improved by rule weighting approach [10,11]. Ishibuchi et al. presented a method for generating fuzzy rules in 1992 [12]. In this method, the sample space is divided into fuzzy subsets. Each subset is expressed by means of the corresponding fuzzy rule. Ishibuchi and Yamamoto calculated the rule weight of fuzzy rules by using data mining techniques in 2004 [8]. Confidence and support terms in association rules are used to rule weighting. Ishibuchi and Yamamoto used different types of rule weight for fuzzy rule-based classification system in 2005 [13]. The method, which was developed by Ishibuchi et al. in 2005, has been tested with different data sets and different types of weights and results are compared.

In this study, we generated fuzzy rules via Ishibuchi and Yamamoto [8]. In [8], candidate rule set is created by using confidence and support terms in associate rules. Rule sets are generated by using previously created candidate fuzzy rule sets' selection criteria. The generated fuzzy rules are weighted using the Gravitational Search algorithm (GSA). GSA is a new algorithm for optimization problems. GSA is an efficient optimization algorithm for huge optimization problems based on Newton's law of gravitation and motion. In [13] four different fuzzy rule weighting methods are introduced and the proposed method is tested against these methods. The proposed method is tested with six different data sets. The simulation results are compared with the results of the studies by Ishibuchi et al. [13].

2. **Fuzzy Rule-Based Classification Systems.** Fuzzy rule-based classification system consists of a database, rule base and reasoning method. The database includes fuzzy sets, and linguistic terms. The rule base composed of fuzzy rules corresponding to the fuzzy subsets. The reasoning method is a mechanism to classify new samples by using the database and the fuzzy rules [14]. The type of fuzzy rules used in this study is expressed as follows.

$$\text{Rule } R_q: \text{ If } x_1 \text{ is } A_{q1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \quad q = 1, 2, \ldots, Q \tag{1}$$

where R_q is the label of the qth fuzzy if-then rule, $x = (x_1, x_2, \ldots, x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set, C_q is a consequent class and CF_q is a rule weight. CF_q is a value between 0 and 1. Q is the total number of fuzzy if-then rules in the rule base.

Two different reasoning methods in [10] were used in this study. These are "weighted vote method" (WVM) and "singles winner method" (SWM). New pattern x_p is classified as class C_w, which is the consequent class of the winner rule R_w in the single winner method. If more than one rule have the same maximum value and different consequent class, the classifier will refuse to classify the new instance. The winner rule is expressed in Equations (2) and (3):

$$\mu_{A_w}(x_p).CF_w = \max\{\mu_{Aq}(x_p).CF_q | R_q \in S\} \tag{2}$$

$$\mu_{Aq}(x_p).CF_w = \mu_{Aq1}(x_{p1}) \times \ldots \times \mu_{Aqn}(x_{pn}) \tag{3}$$

where $\mu_{Aqi}(x_p)$ is the membership function of the antecedent fuzzy set A_{qi}, S is set of fuzzy rules in the fuzzy rule-based classification system.

In the weighted vote method, each fuzzy rule gives a vote for its consequent class and the total strength of votes for each class is calculated. New pattern x_p is classified as the class with the maximum total strength of the vote. The total strength of vote for each class is calculated by Equation (4):

$$V_{Class\ h}(x_p) = \sum_{\substack{R_q \in S \\ C_q = h}} \mu_{Aq}(x_p).CF_q, \quad h = 1, 2, \ldots, M \tag{4}$$

where $\mu_{Aq}(x_p)$ is the membership function of the antecedent fuzzy set A_q, S is set of fuzzy rules in the fuzzy rule-based classification system, M is the number of pattern class.

3. **Generating Fuzzy Rules.** We utilized fuzzy rules generation method of Ishibuchi et al. [4]. According to this method, it is assumed that m labeled patterns $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ are given from M classes for an n-dimensional classification problem. To create fuzzy rules, pattern space is separated into fuzzy subsets. We use the 14 triangular membership functions illustrated in Figure 1. For an n-dimensional problem, the total number of combinations antecedent of fuzzy rules is 14^n. When n is large, the number of fuzzy rules is huge. In this case, it is impossible to use all the fuzzy rules. To overcome this computational load, "don't care" ($\mu_{don't\ care}(x) = 1$) fuzzy set is used. Short-length rules can be created using "don't care". The length of the rule (L) is defined by the number of conditions excluding "don't care".

The concepts of confidence and support in association rules are used for determining consequent class of fuzzy rule, and selecting fuzzy rules. Fuzzy rule R_q in (1) can be

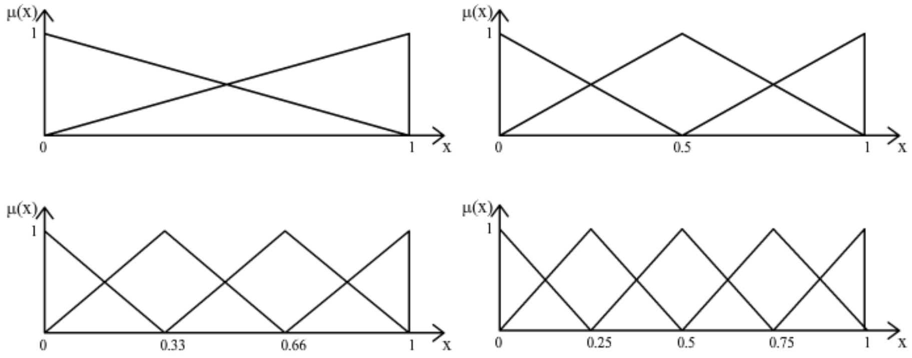

FIGURE 1. 14 triangular membership functions

viewed as an association rule $A_q \Rightarrow C_q$, where A_q is the antecedent conditions of fuzzy rule R_q and C_q is consequent class of fuzzy rule R_q. Confidence and support of fuzzy rule R_q are denoted by c and s, respectively. Confidence and support are defined as they are done by Equations (5) and (6) in [15]:

$$c(A_q \Rightarrow C_q) = \frac{\sum\limits_{x_p \in Class\ C_q} \mu_{Aq}(x_p)}{\sum\limits_{p=1}^{m} \mu_{Aq}(x_p)} \tag{5}$$

$$s(A_q \Rightarrow C_q) = \frac{\sum\limits_{x_p \in Class\ C_q} \mu_{Aq}(x_p)}{m} \tag{6}$$

The confidence is used for finding the consequent class of the fuzzy rule. Confidence values of the fuzzy rule are calculated for each class. The class which has maximum confidence is determined as the consequent class for the fuzzy rule. When the confidence value of each class is the same, the consequent class cannot be determined; and therefore, the fuzzy rule is not generated. Maximum confidence is defined by Equation (7):

$$c(A_q \Rightarrow C_q) = \max(c(A_q \Rightarrow Class\ h) | h = 1, 2, \ldots, M) \tag{7}$$

The generated rules are divided into M groups according to the consequent classes. The fuzzy rules are sorted in ascending order by using selection criteria for each group. The first N rules are selected from M group. Fuzzy rules are generated by chosen $M \times N$ rules.

4. **Learning Rule Weight in FRBCSs.** Our proposed method is compared with Ishibuchi and Yamamoto [13]. In Ishibuchi and Yamamoto [13], the product of confident value and support value is used as the rule selection criterion. In this method, four different weight types were presented. These types of weight are expressed in Equations (8)-(11), respectively:

$$CF_q^{I} = c(A_q \Rightarrow C_q) \tag{8}$$

$$CF_q^{II} = c(A_q \Rightarrow C_q) - \frac{1}{M-1} \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(A_q \Rightarrow Class\ h) \tag{9}$$

$$CF_q^{III} = c(A_q \Rightarrow C_q) - \max\{c(A_q \Rightarrow Class\ h) | h = 1, 2, \ldots, M; h \neq C_q\} \tag{10}$$

$$CF_q^{VI} = c(A_q \Rightarrow C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(A_q \Rightarrow Class\ h) \tag{11}$$

5. **Gravitational Search Algorithm.** GSA is a new algorithm used to solve high-dimensional optimization problems. It was introduced by Rashedi et al. in 2009 [16]. GSA is an algorithm based on Newton's law of gravitation and motion. According to Newton's law of gravitation, particles attract each other in space. This force is proportional to product of their mass and inversely proportional to the square of the distance between them. Newton's law of gravitation is expressed in Equation (12) in [17]:

$$F = G\frac{M_1 \times M_2}{R^2} \tag{12}$$

where F is gravitational force, G is gravitational constant, M is the mass of the particle, R is the distance between particles. According to Newton's law of motion, when a force affects a particle, the particle accelerates, depending on the amount of the force and its mass. Newton's law of motion is expressed in Equation (13) in [17]:

$$a = \frac{F}{M} \tag{13}$$

where a is acceleration.

GSA is a population-based optimization algorithm. Each individual in the population is referred to as an agent. Agents in the system correspond to different solutions. An agent is defined by position and mass values. The mass of the agent is fitness value which is calculated by using position value. Agents change position depending on the force of attraction. The algorithm searches the values of the position with the best fitness value. Consider an optimization problem of d-dimension and N agents. An agent is expressed in Equation (14):

$$X_i = (x_i^1, x_i^2, \ldots, x_i^d,) \text{ for } i = 1, 2, \ldots, N \tag{14}$$

The force between agents i and j at time t and at d dimension is calculated by Equation (15):

$$F_{ij}^d = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}(x_j^d(t) - x_i^d(t)), \quad t = 1, 2, \ldots, T \tag{15}$$

where $M_i(t)$ is mass of agent i at time t. Gravitational constant $G(t)$ is a value decreasing during the search process; $R_{ij}(t)$ is distance between agent i and j. ε is a small constant. T is the total number of iteration. The total force acting agent i is defined by Equation (16):

$$F_i^d(t) = \sum_{\substack{j=1 \\ j \neq i}}^{N} rand_j F_{ij}^d(t) \tag{16}$$

where $rand_j$ is a random number in the interval $[0, 1]$. The mass of agent i at time t is calculated by Equations (17) and (18).

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{17}$$

$$M_i(t) = \frac{m_i(t)}{\sum\limits_{j=1}^{N} m_j(t)} \tag{18}$$

where $fit_i(t)$ is fitness value of agent i at time, $best(t)$ and $worst(t)$ are the best fitness value and the worst fitness value of agent i at time t. The acceleration of agent i at time t and at d dimension is calculated by Equation (19).

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{19}$$

Position ($x_i^d(t)$) and velocity ($v_i^d(t)$) values are updated by using value of acceleration. Position and velocity values are updated with Equations (20) and (21).

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{20}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{21}$$

In accordance with the above definitions, GSA algorithm steps are listed below [16].

1. Search space definition.
2. Create random initial values.
3. Calculate fitness values of all agents.
4. Calculate $G(t)$, $best(t)$, $worst(t)$ and $M_i(t)$ values for all agents.
5. Calculate total forces acting on each agent.
6. Calculate acceleration and velocity values.
7. Update position values for each agent.
8. Repeat steps from 3 to 7 until the stopping criterion has been reached.

6. **Applying GSA in Learning Rule Weight.** In this study, the weights of fuzzy rules in fuzzy rule-based classifier are found by using GSA. The goal of the GSA is to find rule weight values which maximize classification accuracy. Fuzzy rules were generated by the method mentioned in Section 4. The dimension of agents in GSA is the number of rules in fuzzy rule-based classifiers. The classification rate of fuzzy rule-based classifier is used as fitness function. The success rate of classifying test data is calculated by testing obtained weights values on each iteration. Weight values which obtained the highest classification success rate are presented as a solution. The rule is redundant if the weight of any rule is zero. The redundant rules are removed from the rule set. Our method involves the following steps.

1. Generating candidate fuzzy rules.
2. Selecting N fuzzy rules for each consequent class according to selection criterion.
3. Weighting the selected fuzzy rules by using GSA.
4. Creating fuzzy rule-based classifier with weighted fuzzy rules.

The parameters of GAS is selected as 20 agent, 200 total iteration and the change of gravitational constant ($G(t)$) is calculated by Equation (22) for the proposed method.

$$G(t) = G_0 e^{-a\frac{t}{T}} \tag{22}$$

where $G(t)$ is gravitational constant at time t, G_0 is initial gravitational constant, T is the total number of iteration. G_0 is set to 100. a is set to 20. The dimensions of agents are determined by total number of rules generated for classifying all patterns.

7. **Computer Simulation.** In this study, the proposed method is compared with the method described in Section 4. We used wine, glass, sonar, iris, pima indians diabetes and image segmentation data sets for comparisons. These data sets are obtained from UCI machine learning repository. The properties of data sets are shown in Table 1.

Firstly, the features of the data sets are normalized to lie between 0 and 1. The average classification rate is calculated by using leave-one-out ($LV1$) technique. In $LV1$ technique, a single pattern is used as the test data and the other patterns are used as training data. Fuzzy rule-based classifier is designed by using training data and predictive accuracy is evaluated by test data. This process is repeated so that all patterns are used as test data.

14 membership functions illustrated in Figure 1 are used for each feature in the data sets. When the number of attributes increases, the number of possible antecedent conditions will increase exponentially. In practice, it is not possible to work with a huge number of antecedent conditions [8]. Thus, short-length fuzzy rules can be created by using "don't

Table 1. Properties of data sets

Data Set	Number of Attributes	Number of Samples	Number of Classes
Wine	13	178	3
Glass	9	214	6
Sonar	60	208	2
Iris	4	150	3
Pima Indians Diabetes	8	768	2
Image Segmentation	19	2310	7

care" membership function. The length of a fuzzy rule is defined by the number of antecedent conditions excluding "don't care" [13,14]. We generated fuzzy rule of length three or less for wine, glass, iris and image segmentation data sets. The number of features in sonar and pima indians diabetes data sets are huge, so we generated fuzzy rule of length 1 and 2 for the sonar and pima indians diabetes data sets, respectively. The consequent classes of obtained fuzzy rules are determined. Fuzzy rules are divided into groups according to the consequent class (M). Wine, glass, sonar, iris, pima indians diabetes and image segmentation data sets are divided into 3, 6, 2, 3, 2 and 7 groups, respectively. The value of product of confidence and support is used as the rule selection criterion. The fuzzy rule set is generated with N rule selected from each group according to the selection criterion. The fuzzy rule set contains $N \times M$ fuzzy rules. In this study, experiments are carried out by giving different values to N such as 1, 2, 3, 4 and 5. All the test results are obtained by the average value of 30 independent runs.

Classification accuracy of the method by Ishibuchi et al. and our proposed method on wine, glass, sonar, iris, pima indians diabetes and image segmentation data sets are compared. The results of our method and the method by Ishibuchi et al. for wine, glass, sonar, iris, pima indians diabetes and image segmentation data sets are shown by using single winner and weighted vote methods in Table 2, Table 3, Table 4, Table 5, Table 6 and Table 7, respectively.

The results of type 2, type 3 and type 4 in sonar and pima indians diabetes data sets (Table 4 and Table 6) are the same because the number of class in sonar and pima indians diabetes data sets is 2. When number of class of data set is two, the value of type 2, type 3 and type 4 weights are the same.

Table 2. Simulation result of wine data by single winner method and weighted vote method

Reasoning Method	Number of Rules	No rule weights	Type 1	Type 2	Type 3	Type 4	Our Method (average length of rules)
SWM	3	89, 89	89, 89	89, 89	89, 33	89, 33	**91, 57** $(2,70)$
	6	80, 34	83, 15	85, 96	84, 83	85, 39	**96, 63** $(5,45)$
	9	88, 76	91, 57	92, 13	93, 26	93, 26	**99, 44** $(7,94)$
	12	93, 26	93, 26	92, 70	93, 26	93, 26	**100, 00** $(11,19)$
	15	88, 76	91, 57	91, 57	94, 38	93, 26	**99, 44** $(13,54)$
WVM	3	89, 89	89, 89	89, 89	89, 33	89, 33	**92, 13** $(2,79)$
	6	87, 08	87, 64	88, 76	89, 33	88, 76	**96, 63** $(5,49)$
	9	93, 82	93, 26	93, 26	94, 38	93, 82	**100, 00** $(8,35)$
	12	94, 38	94, 94	94, 38	94, 38	93, 26	**99, 44** $(10,90)$
	15	95, 51	95, 51	94, 38	94, 38	93, 82	**100, 00** $(13,72)$

Table 3. Simulation result of glass data by single winner method and weighted vote method

Reasoning Method	Number of Rules	No rule weights	Type 1	Type 2	Type 3	Type 4	Our Method (average length of rules)
SWM	6	45, 79	49, 53	45, 79	39, 25	58, 88	**75, 70** (5,55)
	12	45, 33	48, 60	45, 79	39, 72	67, 76	**73, 36** (11,30)
	18	45, 33	48, 60	45, 79	39, 72	66, 82	**70, 56** (16,77)
	24	45, 33	48, 60	45, 33	40, 19	65, 89	**68, 22** (22,98)
	30	39, 72	48, 13	45, 33	40, 19	54, 21	**65, 42** (29,22)
WVM	6	45, 79	49, 53	45, 79	39, 25	58, 88	**74, 77** (5,53)
	12	45, 33	48, 60	46, 26	39, 25	67, 76	**71, 96** (11,02)
	18	45, 33	47, 20	47, 20	40, 19	68, 22	**69, 63** (16,38)
	24	45, 33	47, 20	48, 60	40, 19	**68, 22**	66,36 (22,54)
	30	45, 79	46, 73	47, 20	42, 06	**66, 36**	65, 89 (29,39)

Table 4. Simulation result of sonar data by single winner method and weighted vote method

Reasoning Method	Number of Rules	No rule weights	Type 1	Type 2	Type 3	Type 4	Our Method (average length of rules)
SWM	2	53, 37	53, 37	53, 37	53, 37	53, 37	**97, 12** (1,75)
	4	52, 88	53, 37	53, 37	53, 37	53, 37	**93, 27** (3,57)
	6	52, 88	53, 37	53, 37	53, 37	53, 37	**91, 35** (5,26)
	8	52, 40	53, 37	53, 37	53, 37	53, 37	**92, 79** (7,34)
	10	52, 40	53, 37	53, 37	53, 37	53, 37	**88, 46** (9,23)
WVM	2	53, 37	53, 37	53, 37	53, 37	53, 37	**95, 67** (1,73)
	4	53, 85	52, 88	53, 37	53, 37	53, 37	**96, 63** (3,44)
	6	51, 92	51, 92	53, 37	53, 37	53, 37	**97, 60** (5,31)
	8	51, 92	50, 48	53, 37	53, 37	53, 37	**95, 19** (7,36)
	10	47, 60	50, 48	53, 37	53, 37	53, 37	**94, 23** (9,08)

Table 5. Simulation result of iris data by single winner method and weighted vote method

Reasoning Method	Number of Rules	No rule weights	Type 1	Type 2	Type 3	Type 4	Our Method (average length of rules)
SWM	3	96, 00	95, 33	95, 33	95, 33	95, 33	**99, 33** (2,81)
	6	92, 67	95, 33	95, 33	95, 33	95, 33	**98, 00** (5,85)
	9	93, 33	95, 33	95, 33	95, 33	95, 33	**98, 67** (8,58)
	12	91, 33	95, 33	95, 33	95, 33	95, 33	**98, 67** (11,75)
	15	88, 67	95, 33	95, 33	95, 33	95, 33	**98, 00** (14,82)
WVM	3	96, 00	95, 33	95, 33	95, 33	96, 00	**100, 00** (2,96)
	6	95, 33	95, 33	94, 67	94, 67	95, 33	**100, 00** (5,74)
	9	96, 00	95, 33	95, 33	95, 33	95, 33	**99, 33** (8,68)
	12	97, 33	95, 33	96, 00	96, 00	95, 33	**98, 67** (11,72)
	15	96, 00	95, 33	95, 33	95, 33	95, 33	**98, 00** (14,54)

In Table 8, best results that obtained from the method of Ishibuchi and the proposed method are compared in terms of classification accuracy and rule counts. These comparisons are compared for both of the reasoning methods SWM and WVM.

The results from Table 8 show that our method has better results in all of the tests. The results in Table 8 show that proposed method increases the classification accuracy for wine, glass, sonar, iris, pima indians diabetes and image segmentation, 5.06%, 7.25%, 43.99%,

Table 6. Simulation result of Pima Indians Diabetes data by single winner method and weighted vote method

Reasoning Method	Number of Rules	No rule weights	Type 1	Type 2	Type 3	Type 4	Our Method (average length of rules)
SWM	2	65, 10	65, 10	65, 10	65, 10	65, 10	**76, 30** (1, 99)
	4	34, 38	65, 10	65, 10	65, 10	65, 10	**74, 35** (3, 99)
	6	34, 38	65, 10	65, 10	65, 10	65, 10	**73, 96** (5, 98)
	8	34, 38	65, 10	65, 10	65, 10	65, 10	**71, 61** (7, 99)
	10	31, 38	66, 15	69, 40	69, 40	69, 40	**71, 48** (9, 99)
WVM	2	65, 10	65, 10	65, 10	65, 10	65, 10	**75, 13** (1, 99)
	4	66, 02	65, 23	64, 97	64, 97	64, 97	**75, 52** (3, 98)
	6	70, 18	68, 23	64, 97	64, 97	64, 97	**74, 48** (5, 98)
	8	68, 88	67, 32	64, 97	64, 97	64, 97	**73, 70** (7, 97)
	10	68, 49	67, 84	65, 36	65, 36	65, 36	**71, 22** (9, 99)

Table 7. Simulation result of Image Segmentation data by single winner method and weighted vote method

Reasoning Method	Number of Rules	No rule weights	Type 1	Type 2	Type 3	Type 4	Our Method (average length of rules)
SWM	7	85, 63	85, 67	85, 58	85, 89	81, 30	**86, 88** (7)
	14	61, 65	71, 86	71, 73	72, 21	69, 44	**87, 19** (14)
	21	67, 49	75, 84	76, 32	76, 10	77, 14	**85, 63** (21)
	28	63, 90	70, 65	70, 48	70, 87	67, 92	**85, 20** (28)
	35	66, 80	71, 17	70, 91	71, 52	68, 35	**84, 90** (35)
WVM	7	85, 63	85, 67	85, 58	85, 89	81, 30	**86, 49** (7)
	14	85, 67	84, 94	84, 72	84, 72	83, 30	**86, 32** (14)
	21	84, 03	83, 33	83, 20	83, 20	82, 08	**84, 55** (21)
	28	83, 12	82, 68	82, 55	82, 81	82, 16	**84, 23** (28)
	35	82, 60	82, 55	82, 47	82, 25	81, 99	**84, 20** (35)

Table 8. Comparision of the best results

Data Set	Reasoning Method	The Best Result of Ishibuchi	The Best Result of Our Method
Wine	SWM	94, 38 (15)	100, 00 (11,19)
	WVM	95, 51 (15)	100, 00 (8,35)
Glass	SWM	67, 76 (12)	75, 70 (5,55)
	WVM	68, 22 (24)	74, 77 (5,53)
Sonar	SWM	53, 37 (2)	97, 12 (1,75)
	WVM	53, 37 (2)	97, 60 (5,31)
Iris	SWM	96, 00 (3)	99, 33 (2,81)
	WVM	97, 33 (12)	100, 00 (2,96)
Pima Indians Diabetes	SWM	69, 40 (10)	76, 30 (1,99)
	WVM	68, 88 (8)	75, 52 (3,98)
Image Segmentation	SWM	85, 89 (7)	87, 19 (14)
	WVM	85, 89 (7)	86, 49 (7)

3%, 6.77% ve 0.95% respectively. As well as improvement in classification accuracy, rule lengths are also decreased. The number of fuzzy rules is reduced a little by the proposed method. However, calculating weights of fuzzy rules brings an additional calculation cost in the proposed method. This situation poses a disadvantage for our proposed method.

Considering the results in terms of classification rate, our method seems to be highly successful. GSA appears to be an effective method used in the learning weights of fuzzy rules.

8. **Conclusions.** In this study, we proposed GSA algorithm to calculate weights of fuzzy rules for fuzzy rule base classifiers. GSA is an optimization technique used to find optimal weights of the rules which maximize the predictive accuracy of the fuzzy classifier. The proposed method compared with the methods by Ishibuchi et al. with six different data sets from UCI. The results show that the proposed method is considerably good from the point of view of classification accuracy. Also proposed method reduces fuzzy rules partially. For future work, multi-objective optimization techniques will be applied to minimize rule length and amount, while maximizing predictive accuracy.

REFERENCES

[1] J. Casillas, O. Cordon, M. J. Del Jesus and F. Herrera, Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems, *Information Sciences*, vol.136, no.1-4, pp.135-157, 2001.

[2] H. Ishibuchi, T. Murata and I. B. Turksen, Single-objective and two-objective genetic algorithms for selecting linguistic rules for patterns classifications problems, *Fuzzy Sets and Systems*, vol.89, no.2, pp.135-149, 1997.

[3] H. Ishibuchi, K. Nozaki, N. Yamamoto and H. Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms, *IEEE Trans. Fuzzy Systems*, vol.3, no.3, pp.260-270, 1995.

[4] Y. Liu, X. Zhu, J. Zhang and S. Wang, Application of particle swarm optimization algorithm for weighted fuzzy rule-based system, *Proc. of the 30th Annual Conference of the IEEE Industrial Electronics Society*, pp.2188-2191, 2004.

[5] K. Siminski, Rule weights in a neuro-fuzzy system with a hierarchical domain partition, *Int. J. Appl. Math. Comput. Sci.*, vol.20, no.2, pp.337-347, 2010.

[6] V. Uebele, S. Abe and M. S. Lan, A neural-network-based fuzzy classifiers, *IEEE Trans. on Systems, Man and Cybernetics*, vol.25, pp.353-361, 1995.

[7] D. Nauck and R. Krusev, A neuro-fuzzy method to learn fuzzy classification rules from data, *Fuzzy Sets and Systems*, vol.89, pp.277-288, 1999.

[8] H. Ishibuchi and N. Yamamoto, Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, *Fuzzy Sets and Systems*, vol.141, pp.59-88, 2004.

[9] M. De Cock, C. Cornelis and E. E. Kerre, Elicitatition of fuzzy association rules from positive and negative examples, *Fuzzy Sets and Systems*, vol.149, no.1, pp.73-85, 2005.

[10] H. Ishibuchi, T. Nakashima and T. Morisawa, Voting in fuzzy rule-based systems or pattern classification problems, *Fuzzy Sets and Systems*, vol.103, pp.223-238, 1999.

[11] H. Ishibuchi and T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, *IEEE Trans. Fuzzy Systems*, vol.9, no.4, pp.506-515, 2002.

[12] H. Ishibuchi, K. Nozaki and H. Tanaka, Distributed representation of fuzzy rules and its application to pattern classification, *Fuzzy Sets and Systems*, vol.52, pp.21-32, 1992.

[13] H. Ishibuchi and N. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, *IEEE Trans. Fuzzy Systems*, vol.13, no.4, pp.428-435, 2005.

[14] O. Cordon, M. J. del Jesus and F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, *International Journal of Approximate Reasoning*, vol.20, pp.21-45, 1999.

[15] H. Ishibuchi, N. Yamamoto and T. Nakashima, Fuzzy data mining: Effect of fuzzy discretization, *Proc. of the 1st IEEE International Conference on Data Mining*, pp.241-248, 2001.

[16] E. Rashedi, H. Nezamandi-Pour and S. Saryazdi, *GSA: A Gravitational Search Algorithm, Information Sciences*, vol.179, pp.2232-2248, 2009.

[17] D. Holliday, R. Resnick and J. Walker, *Fundamentals of Physics*, John Wiley and Sons, 1993.