# Ensemble Fuzzy Rule-Based Classifier Design by Parallel Distributed Fuzzy GBML Algorithms

Hisao Ishibuchi, Masakazu Yamane and Yusuke Nojima

Department of Computer Science and Intelligent Systems, Graduate School of Engineering,
Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
{hisaoi@, masakazu.yamane@ci, nojima@}cs.osakafu-u.ac.jp

**Abstract.** We have already proposed an island model for parallel distributed implementation of fuzzy genetics-based machine learning (GBML) algorithms. As in many other island models, a population of individuals is divided into multiple subpopulations. Each subpopulation is assigned to a different island. The main characteristic of our model is that training patterns are also divided into multiple training data subsets. Each subset is assigned to a different island. The assigned subset is used to train the subpopulation in each island. The assignment of the training data subsets is periodically rotated over the islands (e.g., every 100 generations). A migration operation is also periodically used in our model. Our original intention in the use of such an island model was to significantly decrease the computation time of fuzzy GBML algorithms. In this paper, we propose an idea of using our island model for ensemble classifier design. An ensemble classifier is constructed by choosing the best classifier in each island. Since the subpopulation at each island is evolved using a different training data subset, a different classifier may be obtained from each island to construct an ensemble classifier. This suggests a potential ability of our island model as an ensemble classifier design tool. However, the diversity of the obtained classifiers from multiple islands seems to be decreased by frequent training data subset rotation and frequent migration. In this paper, we examine the effects of training data subset rotation and migration on the performance of designed ensemble classifiers through computational experiments.

**Keywords:** Genetics-based machine learning (GBML), genetic fuzzy systems, fuzzy rule-based classifiers, parallel evolutionary algorithms, island model.

## 1 Introduction

Since the early 1990s [1]-[4], evolutionary algorithms have been frequently used for fuzzy system design. Such a hybrid approach is referred to as a genetic fuzzy system (GFS [5]-[8]). This is because genetic algorithms have been mainly used for fuzzy system design. Multi-objective evolutionary algorithms have also been used for fuzzy system design [9]-[12]. Such a multi-objective hybrid approach is often referred to as a multi-objective genetic fuzzy system (MoGFS).

Applications of evolutionary algorithms to machine learning are called genetics-based machine learning (GBML). Many GBML algorithms have been proposed for rule-based classifier design [13]-[16]. Fuzzy GBML is GBML for fuzzy rule-based classifier design [17]-[20], which can be viewed as a special class of GFS.

GBML algorithms are often categorized into three classes: Pittsburgh, Michigan and iterative rule learning (IRL) approaches [16]. In Pittsburgh approach, an entire rule-based classifier is coded as an individual. As a result, a population of individuals is a set of rule-based classifiers. Since the fitness of each individual is directly related to the performance of the corresponding rule-based classifier, Pittsburgh approach can directly maximize the performance of rule-based classifiers through the search for individuals with high fitness values. In Michigan approach, a single rule is coded as an individual. A population of individuals is handled as a rule-based classifier. Thus Michigan approach indirectly maximizes the performance of rule-based classifiers through the search for good rules with high fitness values. In IRL approach, a single rule is coded as an individual as in Michigan approach. A single rule is obtained from a single run of a GBML algorithm in IRL approach. Thus multiple runs are needed to design a rule-based classifier. Since IRL approach searches for a single best rule in each run, classifier optimization is indirectly and sequentially performed.

In order to drastically decrease the computation time of fuzzy GBML algorithms in Pittsburgh approach, we proposed an idea of parallel distributed implementation using an island model [21]-[24]. As in other island models, a population of individuals is divided into multiple subpopulations. Our island model also divides training patterns into multiple training data subsets. In each island of our island model, a subpopulation is evolved using a training data subset. Let $N_{\mathrm{CPU}}$ is the number of available processors for parallel computation. Since both the population size and the training data size are decreased to $1/N_{\mathrm{CPU}}$ of their original size in each island, the speedup by our island model is the order of $1/(N_{\mathrm{CPU}})^2$. This is the main advantage of our island model over other parallel implementation methods with the speedup of the order of $1/N_{\mathrm{CPU}}$.

In this paper, we propose an idea of using our island model for fuzzy rule-based ensemble classifier design. In our former studies [21]-[24], only a single best fuzzy rule-based classifier was selected from all individuals at the final generation. With no increase in computation load, we can choose a single best fuzzy rule-based classifier from each island to construct an ensemble classifier. Since a different training data subset is used in each island, a different classifier is likely to be obtained from each island. We examine the usefulness of our idea through computational experiments.

This paper is organized as follows. First we briefly explain fuzzy rule-based classifiers and fuzzy GBML algorithms in Section 2. In Section 3, we explain our island model for parallel distributed implementation of fuzzy GBML algorithms. Then we report experimental results in Section 4 where a single best classifier and an ensemble classifier are compared with each other. This comparison is performed under various specifications of two important parameters in our island model. One is a training data subset rotation interval. This parameter controls the frequency of the rotation of training data subsets over islands. The other is a migration interval. This parameter controls the frequency of the migration of individuals (i.e., the migration of fuzzy rule-based classifiers) over subpopulations. Frequent rotation and/or frequent migration may decrease the diversity of classifiers in different subpopulations, which may lead to the deterioration in the performance of designed ensemble classifiers. The effects of rotation and migration are examined through computational experiments in Section 4. Finally we conclude this paper in Section 5.

## 2 Fuzzy Rule-Based Classifiers and Fuzzy GBML Algorithms

In this paper, we used the same parallel distributed fuzzy GBML algorithm as in our former study [24]. Our fuzzy GBML algorithm has a framework of Pittsburgh approach (i.e., an individual is a set of fuzzy rules). Michigan approach is used as a kind of local search for each individual. Our fuzzy GBML algorithm can be viewed as a hybrid algorithm for fuzzy rule-based classifier design (see [17] for details).

Let us assume that we have an $n$-dimensional pattern classification problem with $m$ training patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1, 2, ..., m$. We use fuzzy rules of the following type for our classification problem:

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \tag{1}$$

where $R_q$ shows the $q$th fuzzy rule, $A_{qi}$ is an antecedent fuzzy set ($i = 1, 2, ..., n$), $C_q$ is a class label, and $CF_q$ is a rule weight.

Let $S$ be a fuzzy rule-based classifier for our classification problem. The fuzzy rule-based classifier $S$ is a set of fuzzy rules of the type in (1). When a pattern $\mathbf{x}_p$ is to be classified by $S$, a single winner rule is chosen for $\mathbf{x}_p$ from $S$. The selection of the winner rule for $\mathbf{x}_p$ is based on the compatibility grade of each fuzzy rule $R_q$ with $\mathbf{x}_p$ and the rule weight $CF_q$. More specifically, first the product of the compatibility grade and the rule weight is calculated for each fuzzy rule in $S$ as a winner rule selection criterion. Then the fuzzy rule with the maximum product is chosen as the winner rule for $\mathbf{x}_p$, which is assigned to the consequent class of the winner rule.

Since our study on fuzzy rule-based classifiers in the early 1990s [25], the single winner-based fuzzy reasoning method has been frequently used in fuzzy rule-based classifiers together with fuzzy rules of the type in (1). For other types of fuzzy reasoning methods and fuzzy rules for classification problems, see [26], [27].

Our fuzzy GBML algorithm is used to find the best fuzzy rule-based classifier $S$ with respect to the following fitness function (This fitness function is minimized):

$$fitness(S) = w_1 f_1(S) + w_2 f_2(S) + w_3 f_3(S), \tag{2}$$

where $w_1$, $w_2$ and $w_3$ are non-negative weights, and $f_1(S)$, $f_2(S)$ and $f_3(S)$ are the following measures to evaluate the fuzzy rule-based classifier $S$:

$f_1(S)$: The error rate of $S$ on training patterns in percentage,
$f_2(S)$: The number of fuzzy rules in $S$,
$f_3(S)$: The total rule length over fuzzy rules in $S$.

The total rule length is the total number of antecedent conditions of fuzzy rules in $S$. In the fitness function in (2), $f_1(S)$ is an accuracy measure of $S$ while $f_2(S)$ and $f_3(S)$ are complexity measures of $S$. These three measures are combined into a single fitness function in (2). It is possible to use these measures as separate objectives in multi-objective fuzzy GBML algorithms [10]. In this paper, we use the weighted sum fitness function as in our former study [24].

In our fuzzy GBML algorithm, a fuzzy rule-based classifier (i.e., a set of fuzzy rules) is coded as an integer string. More specifically, antecedent fuzzy sets of fuzzy rules in a fuzzy rule-based classifier are represented by an integer string. The string length is not fixed (i.e., variable string length) because the number of fuzzy rules in fuzzy rule-based classifiers is not pre-specified. The consequent class and the rule weight of each fuzzy rule are not coded. This is because they can be easily determined in a heuristic manner from compatible training patterns with each fuzzy rule (e.g., the consequent class is the majority class among the compatible training patterns [28]).

## 3 Island Model for Parallel Distributed Implementation

Island models have been frequently used for the speedup of evolutionary algorithms through parallel implementation [29]-[31]. We also use an island model for parallel distributed implementation of our fuzzy GBML algorithms. As in other island models, a population of individuals (i.e., rule sets) is divided into multiple subpopulations. Each subpopulation is viewed as an island. The number of subpopulations is the same as the number of processors for parallel computation. A single processor is assigned to each subpopulation (i.e., each island). We use a simple migration operation to periodically send a copy of the best rule set in each island to the next one. We assume a ring structure of islands when the migration operation is executed (i.e., when a copy of the best rule set is sent to the next island). The worst rule set in each island is removed just before migration in order to maintain the subpopulation size unchanged. The number of generations between consecutive executions of migration is referred to as a "migration interval". This is an important parameter in our island model.

The main characteristic of our island model is that training patterns are divided into training data subsets. The number of training data subsets is the same as the number of subpopulations (i.e., the number of islands). A different training data subset is assigned to each island. The assigned training data subsets are periodically rotated over the islands with a ring structure. That is, the training data subset assigned to each island is periodically moved to the next island. The number of generations between consecutive executions of rotation is referred to as a "rotation interval". This is also an important parameter in our island model. Our island model is illustrated in Fig. 1 where training data and a population of rule sets are divided into seven subsets.

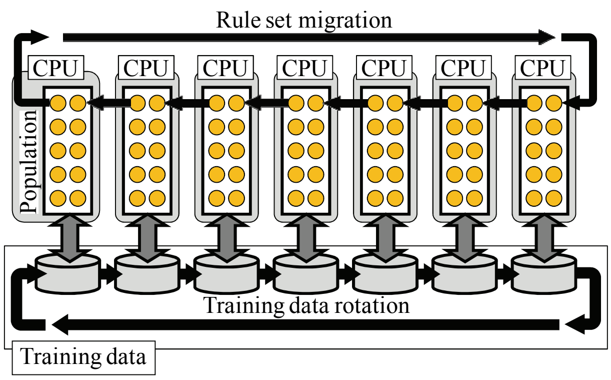



**Fig. 1.** Our island model for parallel distributed implementation [23].

An interesting setting in our island model is that the training data subset rotation is performed in the opposite direction to the migration (see Fig. 1). That is, the training data subset at the $i$th island is moved to the $(i+1)$th island while a copy of the best rule set in the $i$th island is sent to the $(i-1)$th island. This is because the migration severely counteracts positive effects of the training data subset rotation when they are performed in the same direction at the same generation. In this case, a copy of the best rule set is sent to the next island together with the training data subset. For details of negative effects of such a synchronized migration and rotation, see [24].

# 4 Experimental Results

In our computational experiments, we applied our fuzzy GBML algorithm to the satimage data set in the KEEL project database [32]. The satimage data set has 6,435 patterns from six classes. Each pattern has 36 attributes. Since the satimage data set has many attributes (i.e., 36 attributes), we applied our fuzzy GBML algorithm to this data set using a large computation load: 50,000 generations of a population with 210 rule sets. Our computational experiments were performed on a workstation with two Xeon 2.93 GHz quad processors (i.e., eight CPU cores in total). Among the eight CPUs of the workstation, seven CPUs were used for parallel computation. This means that a population of 210 rule sets was divided into seven subpopulations of size 30 in our island model for parallel distributed implementation.

In our computational experiments, the following three variants of our fuzzy GBML algorithm were compared:

1. Standard non-parallel non-distributed model for single classifier design [17]
2. Our island model for single classifier design [23]
3. Our island model for ensemble classifier design

The standard non-parallel non-distributed algorithm was executed on a single CPU of the work station while seven CPUs were used in the two variants of parallel distributed implementation. We used the ten-fold cross-validation procedure (i.e., 10CV) in our computational experiments. The 10CV was iterated three times to calculate the average test data accuracy as well as the average training data accuracy of designed classifiers by each of the above-mentioned three variants (i.e., $3 \times 10$CV was used for performance evaluation in this paper).

In the first two variants for single classifier design, the best rule set with respect to the fitness function in (2) for all training patterns was chosen from the final population. In the last variant for ensemble classifier design, the best rule set with respect to the fitness function in (2) for the assigned training data subset was chosen from each subpopulation. The selected seven rule sets were used as seven fuzzy rule-based classifiers in an ensemble classifier. Pattern classification was performed in the designed ensemble classifier using a simple majority voting scheme as follows: First a pattern was classified by each of the seven fuzzy rule-based classifiers. According to the classification result, each classifier voted for a single class. The final classification result by the ensemble classifier was the class with the maximum vote. When multiple classes had the same maximum vote, one of those classes was randomly chosen.

From the non-parallel non-distributed algorithm, we obtained the following results:

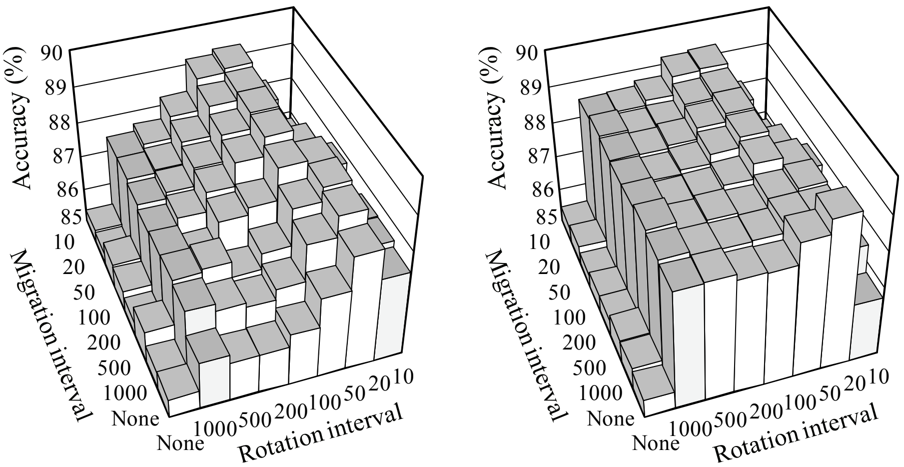
Average classification rate on training data: 86.31%
Average classification rate on test data: 84.46%
Average computation time: 658.89 minutes (10.98 hours) for a single run.

In Fig. 2, we summarize experimental results by our island model on training data. The vertical axis of each plot (i.e., the height of each bar) is the average classification rate on training data. The two axes of the base of each plot show the rotation interval and the migration interval. As shown in Fig. 2, we examined $8 \times 8$ combinations of the following specifications in our computational experiments:

Rotation interval: 10, 20, 50, 100, 200, 500, 1000, None,
Migration interval: 10, 20, 50, 100, 200, 500, 1000, None.

In these specifications, “None” means that we did not use the rotation and/or the migration. For example, when the rotation interval was 10 and the migration interval was “None” (i.e., the bottom-right bar in each plot of Fig. 2), only the training data subset rotation was performed every 10 generations.

From the comparison between the two plots in Fig. 2, we can see that better results with higher training data accuracy were obtained by ensemble classifiers in a wide range of parameter specifications in Fig. 2 (b). For example, average classification rates higher than 88% were always obtained in Fig. 2 (b) from the five specifications of the rotation interval: 50, 100, 200, 500 and 1000 (i.e., $5 \times 8$ combinations). However, such a good result was not obtained in Fig. 2 (a) when the rotation interval was specified as 500 or 1000. When the rotation interval was 1000, the fuzzy GBML algorithm was executed in each island for 1000 generations using the assigned training data subset. In this case, it is likely that each subpopulation was over-fitted to the assigned training data subset during 1000 generations. As a result, any classifiers are not likely to have high accuracy for the entire training data as shown in Fig. 2 (a).




(a) Accuracy of single best classifiers. (b) Accuracy of ensemble classifiers.

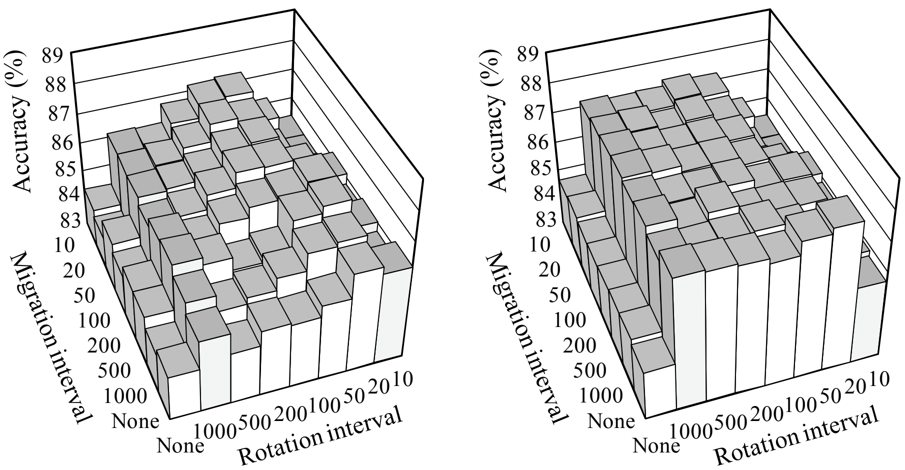
**Fig. 2.** Experimental results of our island model on training data.

Whereas good results were not obtained from over-fitted classifiers in the case of the rotation interval of 1000 in Fig. 2 (a), they can be good components (i.e., good base classifiers) of an ensemble classifier. This is because classifiers in different islands are likely to be over-fitted to different training data subsets. That is, a set of classifiers from different islands is likely to have high diversity. High diversity of base classifiers is essential in the design of high-performance ensemble classifiers. Actually, ensemble classifiers with high accuracy were obtained in Fig. 2 (b) by choosing the locally best classifier with respect to the assigned training data subset from each island when the rotation interval was large (e.g., 500 and 1000).

It should be noted that almost all average classification rates in Fig. 2 (a) are higher than the result 86.31% by the non-parallel non-distributed algorithm (except for the eight combinations with no rotation in Fig. 2 (a)). This observation suggests that the rotation of training data subsets has a positive effect on the search ability of the fuzzy GBML algorithm to find good classifiers. At each island, the training data subset rotation can be viewed as a periodical change of the environment. Such a periodical change seems to help the fuzzy GBML algorithm to escape from local optima. When we did not use the rotation, good results were not obtained in Fig. 2 (a).

In Fig. 2, good results were not obtained from the rotation interval of 10, either. It seems that too frequent changes of the environment had a negative effect on the search ability of the fuzzy GBML algorithm. In this case, the use of ensemble classifiers did not work well. This may be because similar classifiers were obtained from different islands due to the frequent rotation of the assigned training data subsets.

In Fig. 3, we summarize experimental results by our island model on test data. We can obtain almost the same observations from Fig. 3 as those from Fig. 2. That is, better results were obtained from ensemble classifiers in Fig. 3 (b) in a wide range of parameter specifications than single best classifiers in Fig. 3 (a). Especially when the rotation interval was large (e.g., 500 or 1000), good results were obtained from ensemble classifiers whereas the test data accuracy of single classifiers was not high.




(a) Accuracy of single best classifiers. (b) Accuracy of ensemble classifiers.

**Fig. 3.** Experimental results of our island model on test data.

Except for the eight combinations with no rotation in each plot in Fig. 3, almost all average classification rates are higher than the result 84.46% by the non-parallel non-distributed algorithm. This observation in Fig. 3 supports a positive effect of the training data subset rotation on the test data accuracy of obtained fuzzy rule-based classifiers and their ensemble classifiers.

When we used the non-parallel non-distributed algorithm, the average computation time for a single run of our fuzzy GBML algorithm was 658.89 minutes (10.98 hours). The average computation time was drastically decreased by the use of our island model with seven CPUs for parallel computation. We summarize the average computation time of our island model in Fig. 4. The vertical axis (i.e., the height of each bar) shows the average computation time. We can see from Fig. 4 that the average computation time of our fuzzy GBML algorithm was decreased from about 11 hours of the non-parallel non-distributed algorithm to about 20 minutes (about 1/33 of 11 hours) by the use of our island model. In our island model, the size of the assigned subpopulation to each CPU is 1/7 of the population size. Moreover, the size of the assigned training data subset to each CPU is 1/7 of the training data size. As a result, the computation time of our island model can be potentially decreased up to the order of 1/49 from the case of the non-parallel non-distributed algorithm.

It should be noted that there is no difference in the computation time of our island model between single best classifier design and ensemble classifier design. Both a single best classifier and an ensemble classifier are obtained from a single run of our island model. The difference between these two approaches is only the selection of classifiers from the final population after the execution of our island model.

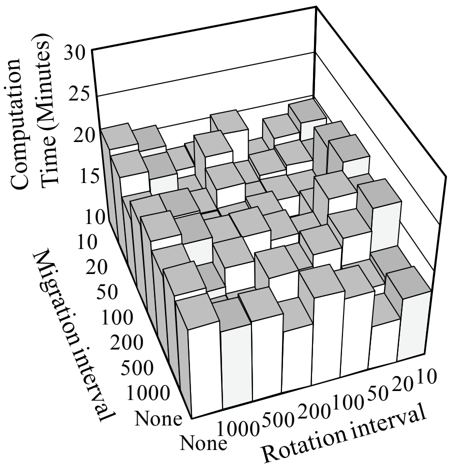



**Fig. 4.** Average computation time of our island model.

## 5 Conclusions

In this paper, we proposed an idea of using an island model for ensemble classifier design. In our island model, a population of classifiers was divided into multiple subpopulations as in other island models. Training patterns were also divided into multiple training data subsets. A different training data subset was assigned to each island (i.e., each subpopulation). The assigned training data subsets were periodically rotated over the islands. After the execution of our island model, the locally best classifier was selected from each island. Through computational experiments, we demonstrated that good ensemble classifiers with high accuracy were obtained from a wide range of parameter specifications of the migration and rotation intervals.

If the generalization ability maximization is the main goal in classifier design, the use of our island model for ensemble classifier design seems to be a good choice. However, if the interpretability of classifiers is also important, ensemble approaches are not recommended. This is because a set of classifiers is usually less interpretable than a single classifier. In this case, we may need a multi-objective approach to find a good accuracy-interpretability tradeoff. Parallel distributed implementation of multi-objective fuzzy GBML algorithms is an interesting future research issue.

Our island model for parallel distributed implementation is a general framework for Pittsburgh-style GBML algorithms. This means that we can use our island model for many other GBML algorithms. Performance evaluation of our island model for other GBML algorithms on other data sets is also an interesting future research issue.

## References


1. Thrift, P.: Fuzzy Logic Synthesis with Genetic Algorithms. Proc. of 4th International Conference on Genetic Algorithms (1991) 509-513
2. Karr, C. L.: Design of an Adaptive Fuzzy Logic Controller using a Genetic Algorithm. Proc. of 4th International Conference on Genetic Algorithms (1991) 450-457
3. Karr, C. L., Gentry, E. J.: Fuzzy Control of pH using Genetic Algorithms. IEEE Trans. on Fuzzy Systems 1 (1993) 46-53
4. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selecting Fuzzy If-Then Rules for Classification Problems using Genetic Algorithms. IEEE Trans. on Fuzzy Systems 3 (1995) 260-270
5. Cordón, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends. Fuzzy Sets and Systems 141 (2004) 5-31
6. Herrera, F.: Genetic Fuzzy Systems: Status, Critical Considerations and Future Directions. International Journal of Computational Intelligence Research 1 (2005) 59-67
7. Herrera, F.: Genetic Fuzzy Systems: Taxonomy, Current Research Trends and Prospects. Evolutionary Intelligence 1 (2008) 27-46
8. Cordón, O.: A Historical Review of Evolutionary Learning Methods for Mamdani-Type Fuzzy Rule-Based Systems: Designing Interpretable Genetic Fuzzy Systems. International Journal of Approximate Reasoning 52 (2011) 894-913
9. Ishibuchi, H., Murata, T., Turksen, I. B.: Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems. Fuzzy Sets and Systems 89 (1997) 135-150
10. Ishibuchi, H., Nakashima, T., Murata, T.: Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction, Information Sciences 136 (2001) 109-133
11. Ishibuchi, H.: Multiobjective Genetic Fuzzy Systems: Review and Future Research Directions (2007) Proc. of 2007 IEEE International Conference on Fuzzy Systems (2007) 913-918

12. Fazzolari, M., Alcalá, R., Nojima, Y., Ishibuchi, H., Herrera, F.: A Review of the Application of Multi-Objective Genetic Fuzzy Systems: Current Status and Further Directions. IEEE Trans. on Fuzzy Systems (to appear)
13. Freitas, A. A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer (2002)
14. Bull, L., Bernado-Mansilla, E., Holmes, J.: Learning Classifier Systems in Data Mining, Springer (2008)
15. García, S., Fernández, A., Luengo, J., Herrera, F.: A Study of Statistical Techniques and Performance Measures for Genetics-Based Machine Learning: Accuracy and Interpretability. Soft Computing 13 (2009) 959-977
16. Fernández, A., García, S., Luengo, J., Bernadó-Mansilla, E., Herrera, F.: Genetics-Based Machine Learning for Rule Induction: State of the Art, Taxonomy, and Comparative Study. IEEE Trans. on Evolutionary Computation 14 (2010) 913-941
17. Ishibuchi, H., Yamamoto, T., Nakashima, T.: Hybridization of Fuzzy GBML Approaches for Pattern Classification Problems. IEEE Trans. on Systems, Man, and Cybernetics - Part B 35 (2005) 359-365
18. Ishibuchi, H., Nojima, Y.: Analysis of Interpretability-Accuracy Tradeoff by Multiobjective Fuzzy Genetics-Based Machine Learning. International Journal of Approximate Reasoning 44 (2007) 4-31
19. Abadeh, M. S., Habibi, J., Lucas, C.: Intrusion Detection using a Fuzzy Genetics-Based Learning Algorithm. Journal of Network and Computer Applications 30 (2007) 414-428
20. Orriols-Puig, A., Casillas, J., Bernadó-Mansilla, E.: Genetic-Based Machine Learning Systems are Competitive for Pattern Recognition. Evolutionary Intelligence 1 (2008), 209-232
21. Nojima, Y., Ishibuchi, H., Kuwajima, I.: Parallel Distributed Genetic Fuzzy Rule Selection. Soft Computing 13 (2009) 511-519
22. Nojima, Y., Mihara, S., Ishibuchi, H.: Parallel Distributed Implementation of Genetics-Based Machine Learning for Fuzzy Classifier Design. Lecture Notes in Computer Science, Vol. 6457: SEAL 2010, Springer, Berlin (2010) 309-318
23. Ishibuchi, H., Mihara, S., Nojima, Y.: Training Data Subdivision and Periodical Rotation in Hybrid Fuzzy Genetics-Based Machine Learning. Proc. of 10th International Conference on Machine Learning and Applications (2011) 229-234
24. Ishibuchi, H., Mihara, S., Nojima, Y.: Parallel Distributed Hybrid Fuzzy GBML Models with Rule Set Migration and Training Data Rotation. IEEE Trans. on Fuzzy Systems (under review)
25. Ishibuchi, H., Nozaki, K., Tanaka, H.: Distributed Representation of Fuzzy Rules and Its Application to Pattern Classification. Fuzzy Sets and Systems 52 (1992) 21-32
26. Cordón, O., del Jesus, M. J., Herrera, F.: A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems. International Journal of Approximate Reasoning 20 (1999) 21-45
27. Ishibuchi, H., Nakashima, T., Morisawa, T.: Voting in Fuzzy Rule-Based Systems for Pattern Classification Problems. Fuzzy Sets and Systems 103 (1999) 223-238
28. Ishibuchi, H., Yamamoto, T.: Rule Weight Specification in Fuzzy Rule-Based Classification Systems. *IEEE Trans. on Fuzzy Systems* 13 (2005) 428-435
29. Alba, E., Tomassini, M.: Parallelism and Evolutionary Algorithms. IEEE Trans. on Evolutionary Computation 6 (2002) 443-462
30. Nedjah, N., Alba, E., de Macedo Mourelle, L.: Parallel Evolutionary Computations. Springer, Berlin (2006)
31. Luque, G., Alba, E.: Parallel Genetic Algorithms: Theory and Real World Applications. Springer, Berlin (2011)
32. KEEL dataset repository, http://keel.es/