

# Recursive Learning of Genetic Algorithm Featuring Incremental Attribute Learning for Higher Dimensional Classification Problems

Haofan Zhang, Sheng-Uei Guan, and Mengjun Xu

**Abstract**—Rule-based classifiers trained by Genetic Algorithms (GAs) have been one of the most prevailing solutions for pattern classification problem. This paper introduces an algorithm named Recursive Learning of Genetic Algorithms featuring Incremental Attribute Learning (RLGA-IAL) developed from the Recursive Learning of Genetic Algorithms with Task Decomposition and Varied Rule Set (RLGA). Instead of training all the attributes in a batch, RLGA-IAL integrates the attributes sequentially. By projecting a large multidimensional search space to single-dimensional spaces with integration, it reduces the difficulty in deriving the classification rules. Though a series of experiments, RLGA-IAL shows a successful and promising performance in classification problems with the dimension of the datasets ranging from 5 to 60.

**Index Terms**—Genetic algorithm, high dimensional classification, incremental attribute learning, recursive learning.

## I. INTRODUCTION

Genetic Algorithms (GAs) have been widely applied in the realm of computer science to deal with various challenging tasks among which pattern classification problems triggered researchers' enormous interests. As a result, many successful GA-based classifiers have been designed and applied in a very broad context. Nevertheless, GA-based classifiers generally suffer from drawbacks regarding classification accuracy and efficiency. Therefore, the *Recursive Learning of Genetic Algorithms with Task Decomposition and Varied Rule Set* (RLGA) was proposed to address these issues [8]. RLGA has gained significant improvement for the training of data under a dimension of 15 but still suffers from degradation when it encounters higher order dimensional problems.

Aiming at resolving the above-mentioned problem, this paper proposes a new algorithm framework, i.e. the *Recursive Learning of Genetic Algorithm featuring Incremental Attribute Learning* (RLGA-IAL) to tackle the performance issues of previous GA-based classifiers in higher dimensional classification problems (with dimension covering 20-60). By applying Incremental Attribute Learning (IAL), the attributes are incrementally trained and integrated

other than learnt in a batch. Hence RLGA-IAL is enabled with the capability to decompose the task more effectively with an improved performance.

To give a comprehensive and objective evaluation on our proposed algorithm, a series of experiments were conducted on a number of benchmark datasets to compare the performance of RLGA-IAL, RLGA and canonical GAs; and the results indicates that RLGA-IAL can handle higher dimensional problems more effectively than the other GA-based approaches discussed in this paper.

## II. RELATED WORK

Extensive research has been conducted in the past decade on genetic algorithms and their applications in pattern classification problems. In this section, a brief review of canonical GA-based classifiers, task decomposition techniques and the original RLGA will be given.

### A. Genetic Algorithms Based Classifiers

As one of the major subfields of computational intelligence, evolutionary computation with its outstanding capability draws enormous attention from researchers, among which Genetic Algorithms (GAs) are especially prevailing and widely applied in searching and optimization problems [10]. One particular application of GAs is on the training of rule-based classifiers for pattern classification problems, and the most successful attempts on various applications in history are the Pittsburgh approach and Michigan approach [6].

Nevertheless, the merits of GA also coexist with many demerits, for example, early convergence. Moreover, comparing with artificial neural networks, naive Bayes classifier, decision trees, supportive vector machines, and other popular machine learning algorithms [1][5][15][17], traditional GA classifier have their weaknesses in efficiency and classification accuracy, especially when confronting higher dimensional problems.

### B. Task Decomposition

The issue of task decomposition has been discussed in the past decades from various perspectives. Most common approaches for task decomposition include domain decomposition [3],[16], input decomposition[9][11]-[13], class decomposition [2][4][14] and hybrid decomposition approach [18] which have all gained various achievements on different applications. Nearly all decomposition approaches are imposed manually by humans' heuristics rather than being done intelligently and automatically according to specific problem context. RLGA was thereby proposed to

Manuscript received September 26, 2012; revised November 15, 2012.

Haofan Zhang is with University of Waterloo, Canada (e-mail: haofan.zhang08@student.uwaterloo.ca).

Sheng-Uei Guan is with the Computer Science and Software Engineering Department at Xi'an Jiaotong-Liverpool University, Suzhou, China (e-mail: steven.guan@xjtlu.edu.cn).

Mengjun Xu is with Xi'an Jiaotong-Liverpool University, Suzhou, China (e-mail: mengjun.xu09@student.xjtlu.edu.cn).

deal with this problem [8].

### C. Recursive Learning of Genetic Algorithms

To address the performance issues of GA based classifiers in terms of both classification accuracy and efficiency, RLGA was proposed in [8]. One particular innovative feature is the elicitation of local fitness. Traditional Pittsburgh GA classifiers generally define their fitness (global fitness) of a chromosome on the view of whole search space whereas the local fitness is defined as the fitness of a chromosome in a local region of search space. It automatically decomposes the learning data through applying a number of sub-processes to study the partitioned training data. The experimental results in [8] illustrated the improvement of RLGA over traditional GA classifiers.

## III. DESIGN OF INCREMENTAL ATTRIBUTE LEARNING

### A. Encoding Mechanism

The encoding mechanism of RLGA-IAL is inherited from the one in RLGA. A chromosome generally encodes a set of rules in Pittsburgh approach, however, in RLGA-IAL, each chromosome encodes only one classification rule which is represented as an IF-ELSE clause. For example, rule:

$$IF(V_{1min} \leq x_1 \leq V_{1max}) \wedge (V_{2min} \leq x_2 \leq V_{2max}) \dots \wedge (V_{nmin} \leq x_n \leq V_{nmax}) THEN y = C \quad (1)$$

The typical encoding of above rule in RLGA (and other Pitts GAs Classifiers) is fully described in Table I.

TABLE I: ENCODING MECHANISM FOR CLASSIFICATION RULE

Antecedent Gene 1	...	Antecedent Gene n	Consequence Gene
Act <sub>1</sub>	V <sub>1min</sub> V <sub>1max</sub>	Act <sub>n</sub> V <sub>nmin</sub> V <sub>nmax</sub>	C

### B. Global Fitness, Local Fitness and Coverage

Normal rule-based classifiers trained by GA generally use the following fitness function, and this is what we named as Global Fitness:

$$f_{Global} = \frac{C}{N_{Total}} = \frac{\text{Number of Correctly Classified Patterns}}{\text{Total Number of Patterns}} \quad (2)$$

RLGA uses the idea of Local Fitness instead:

$$f_{Local} = \frac{C}{N_{Applicable}} = \frac{\text{Number of Correctly Classified Patterns}}{\text{Number of Applicable Patterns}} \quad (3)$$

Apart from Local Fitness, RLGA also introduced a concept of coverage:

$$overage = \frac{N_{Applicable}}{N_{Total}} = \frac{\text{Number of Applicable Patterns}}{\text{Total Number of Patterns}} \quad (4)$$

For the chromosomes that share the same local fitness, we generally prefer the one that covers more patterns, i.e. has a higher coverage.

### C. Integration of New Attributes

Here Fig. 1 gives an intuitive idea of how the integration actually works. Part (a) of the diagram illustrates the process of forming a new rule based on a previously generated rule with an incoming rule whereas Part (b) shows how this

process help to form a new population based on the original population and incoming elements.

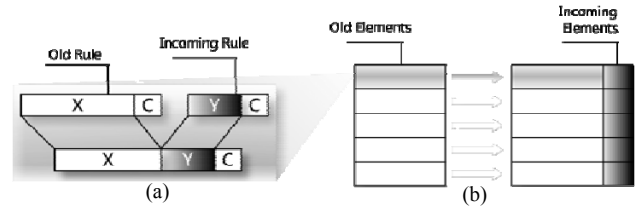


Fig. 1. Formation of a new chromosome and a new population.

### D. Order of Attributes and Discriminating Ability

The order of attributes to be integrated is crucial for RLGA-IAL. To determine the order of the attributes to be added into the chromosomes, we use the idea of Discriminating Ability (DA) [7]. DA is measured by the training classification accuracy achieved by applying Normal GA classifier to the dataset that has only a single active attribute.

## IV. EXPERIMENTS AND ANALYSIS

### A. Cross-Validation

The cross-validation method is applied to test the performance of various algorithms. It focuses on how the classifier model can generalize to an independent data set and thereby can be treated as an accurate indicator to assess how the classifiers will perform in practice. The technique of K-fold cross-validation is the type of cross-validation we utilized in this paper. It first randomly partitions the original datasets into K subsets, among which only one subset is preserved as the testing data set and remaining K-1 are merged into a training data set. This process is repeated K times, with each of the K subsets used exactly once as the testing data. The results of K-fold are finally averaged to produce one final result. We have K = 10 throughout the whole experiment process. Furthermore, each fold has 10 runs and thereby we have 10 × 10 = 100 runs on each benchmark in total. All the experimental results shown in the following sections are taken as the average of these 100 runs of experiments.

### B. Experimental Results

This section presents a series of experimental results, including a full list of classification accuracy with intuitive diagrams and analysis on typical benchmark datasets. All the experiments were conducted on a PC with Intel® Core™2 Duo E7500 CPU 2.93GHz/2.94GHz with 2.0GB RAM and 32-bit Windows 7 Premium Operating System under Weka Software (version 3.6.5). The benchmark datasets used are retrieved from the UCI (University of California, Irvine) Machine Learning Repository.

The experimental results shown in Table II give an overall idea on the performance of three classification algorithms. This table is sorted in ascending order based on the attributes number (also counts in the class attribute). Both Table II and Fig. 2 mainly focused on classification accuracy. Note that, since the parameters are important to the performance of all the classifiers, they have been tuned to reasonable values to deliver relatively good results under their capacity. The

parameters, especially *RuleNumber*, *PopulationSize* are set to a comparable level in order to guarantee the fairness for all algorithms. The features of these classifiers should be taken into consideration as well. For instance, we have *RuleNumber* = 100 & *PopulationSize* = 100 for Normal GA, whereas we need at least *PopulationSize* = 1000 for RLGA because each RLGA chromosome encodes only one rule. However, setting *PopulationSize* = 100 might be sufficient for RLGA-IAL, as it needs to invoke Normal GA when an integration of attribute occurs and this process might generate a number of chromosomes itself.

TABLE II: EXPERIMENTAL RESULTS ON TESTING CLASSIFICATION ACCURACY

Benchmark (Attr. No.)	Normal GA <sup>†</sup>		RLGA <sup>‡</sup>		RLGA-IAL <sup>§</sup>	
	CR <sup>¶</sup>	UR	CR	UR	CR	UR
Iris (5)	93.47	0.00	93.60	0.47	93.73	2.13
Yeast (9)	42.30	0.00	48.15	3.28	41.95	10.78
Glass (10)	58.02	0.05	65.04	1.82	54.29	2.58
Cancer-Wisc (10)	95.15	0.14	95.84	0.01	93.96	0.31
Cancer (10)	69.81	4.67	64.49	0.18	65.57	0.21
Wine (14)	86.85	0.84	91.17	1.53	87.85	1.86
Labor (17)	75.97	1.83	78.40	4.93	72.53	3.60
Vote (17)	61.91	34.29	74.26	21.24	87.87	8.35
Zoo (18)	54.55	25.16	44.51	54.09	93.09	4.04
Lymphography (19)	72.06	1.87	67.58	14.81	77.33	0.62
Vehicle (19)	42.83	1.28	70.14	0.24	58.43	6.00
Hepatitis (20)	76.32	8.31	71.15	11.03	77.07	1.35
Segment (20)	44.33	7.40	96.81	0.17	87.72	7.34
Horse-Colic (23)	64.89	18.70	76.79	0.97	77.28	0.68
Autos (26)	42.50	3.79	60.28	21.49	81.33	1.46
Ionosphere (35)	62.59	24.61	73.17	20.29	85.24	2.40
Anneal (39)	77.31	0.41	0.00	100.00	97.55	0.89
Sonar (61)	26.41	60.46	0.00	100.00	66.47	1.83

¶ CR stands for Classification Rate and UR stands for Unclassified Rate. All of them are in percentage. These notations are the same in the coming sections.

† Mutation Rate = 0.01, Crossover Rate = 1.0, Population Size = 100, Survival Rate = 0.5, Active Rate = 0.5. For Attr No. ≤ 15 we set Rule Number = 30 and for Attr No. > 15 we set Rule Number = 100.

‡ Rule Number Per Expert = 1, Mutation Rate = 0.01, Crossover Rate = 1.0, Survival Rate = 0.5, Active Rate = 0.1. For Attr No. ≤ 15 we set Population Size = 100 and for Attr No. > 15 we set Population Size = 1000

§ Rule Number Per Expert = 1, Mutation Rate = 0.01, Crossover Rate = 1.0, Population Size = 100, Survival Rate = 0.5, Active Rate = 0.1, Attr Order = Descending.

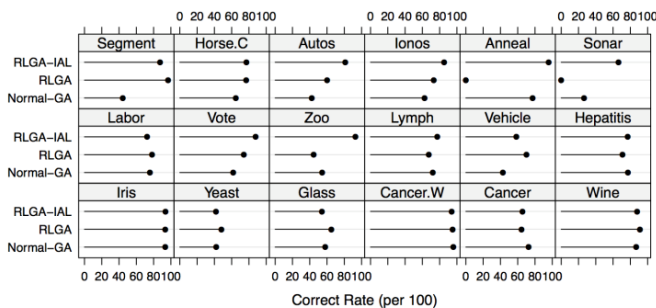


Fig. 2. Classification rate for all datasets.

[8] mentioned the fact that RLGA has better performance compared with normal GA in lower dimensional problems. This point is vetted by our experimental results except for the Cancer datasets. Nevertheless, as the number of attributes increases, its performance is no longer dominant among three.

Meanwhile, the number of instances is another determinant factor on the performance of RLGA. Upon datasets with fewer instances, such as Zoo, RLGA can hardly achieve higher classification accuracy even encompass a reasonable size of population (1000 in this case). Assigning RLGA with a larger population size does not help. For higher order dimensional problems, the advantage of RLGA-IAL becomes obvious.

Generally speaking, the algorithms could be ranked as RLGA-IAL > RLGA > Normal-GA based on an overall estimation of their performance. Nevertheless, there are always some exceptions. For example, the winner on Cancer benchmark is actually the Normal-GA. Note that RLGA's performance on segment dataset is exceptional where RLGA-IAL ranks 2nd.

C. Result Analysis

A brief description of each dataset will be given along with experimental results covering classification accuracy and training time. The classification rate is shown with the SD (standard deviation) value. For simplicity, the Classification Rate, Incorrect Rate, Unclassified Rate and Training Time are abbreviated as CR, IR, UR and Time separately. The rates are all in percentage and the Time is in seconds. Moreover, the *pop* stands for the population size for different RLGA classifier.

1) Ionosphere, anneal, and sonar

Ionosphere, Anneal and Sonar were taken from the group of higher-order dimensional datasets that have attributes number 35, 39 and 61 separately (see Table III). They are the highest dimensional dataset we used as benchmarks and are thereby the major interests of us. RLGA-IAL shows its capacity when the dimension number increases, especially when the number of attributes exceed a limit of 20. Especially for the last two benchmark datasets, while RLGA has degraded to 0, the performance of RLGA-IAL still maintained a relatively good record. In addition, the experimental results show that RLGA's performance greatly relies on the instance number. It also suggests that the number of instances could be very helpful for RLGA to compensate its inability in higher dimensional problem. This also shows that, RLGA-IAL is effective in dealing with situations lacking enough training instances.

2) Lower dimensional data

Another interesting fact we can observe from the experimental results is that RLGA-IAL performs inferior to RLGA in lower dimensional data. There are many possible causes for this. Firstly, the information possessed by each attribute is apparently problem-dependent, and a uniform solution is not necessarily a winner for all. Due to the nature of RLGA-IAL, it gradually decomposes the problem in the incremental learning process. Therefore it is possible that not all attributes are utilized for the classification. This is an inevitable tradeoff if we want to keep the automatic task decomposition feature of RLGA. With instances removed from training set, some useful attributes are even still not integrated. Lower dimensional data tend to possess more information in each attribute, and there are rarely redundant attributes. Accordingly, RLGA-IAL might not be that effective in some lower dimensional problems.

TABLE III: EXPERIMENTAL RESULTS ON IONOSPHERE, ANNEAL & SONAR BENCHMARK

Ionosphere	CR(SD)	IR	UR	Time
Normal-GA	62.59	12.79	24.61	27.71
RLGA (pop=100)	10.53	1.39	88.09	0.02
RLGA (pop=500)	42.78	5.05	52.17	9.32
RLGA (pop=1000)	73.17	6.54	20.29	28.50
RLGA-IAL	85.24	31.70	1.83	99.63
Anneal	CR(SD)	IR	UR	Time
Normal-GA	77.31	22.28	0.41	146.45
RLGA (pop=100)	0.00	0.00	100.00	4.71
RLGA (pop=500)	0.00	0.00	100.00	21.31
RLGA (pop=1000)	9.66	0.34	90.00	29.31
RLGA-IAL	97.55	0.89	1.98	198.93
Sonar	CR(SD)	IR	UR	Time
Normal-GA	26.41	13.13	60.46	20.46
RLGA (pop=100)	0.00	0.00	100.00	0.01
RLGA (pop=500)	0.00	0.00	100.00	0.15
RLGA (pop=1000)	0.00	0.00	100.00	0.3
RLGA-IAL	66.47	31.70	1.83	105.90

### 3) Order of Attributes in Incremental Learning Process

One issue that cannot be overlooked is the ordering of attributes for attribute integration in the incremental learning process. As explained before, the Discriminating Ability (DA) is the criteria to rank the attributes. The default option for ordering attributes is based on the descending order of DA, that is, we integrate attribute from the one with the highest DA to the one with the lowest.

Although previous research has been conducted to show the descending order is best under IAL [9], whether it is the case for RLGA remains an open question. Since the theoretical foundation is not there, we need to conduct a series of experiments. Here we take two benchmarks, i.e. Anneal and Segment, for testing. The experimental results focus on CR, UR and training Time. Other than descending order, the orders that are also investigated include ascending order (reverse of descending order), natural order (original attribute order) and random order.

Intuitively, descending order should have the best performance, because the attribute that has higher DA can discriminate the instances better. Ideally, we can classify with fewer attributes rather than the full attribute set and most possible subset of attributes are those that possess higher DAs. When a portion of instances are perfectly learnt by the classifier, the instances can be removed from original training set and thereby simplifying the classification task as a whole. With these considerations in mind, we will see how the number of remaining patterns changes along with the attribute integration process.

Both results from Anneal and Segment in Table IV show the “Descending Order” is the best among all four ways. They result in best classification rate with minimum training time. “Ascending Order”, on the opposite, always delivers the worst result. The random order and natural order generally lie between ascending and descending, but they do not give much correlative information. This is in accord with our previously mentioned point of view that it is actually the information held by each attribute hold determines its usefulness. The experiments are conducted without specific domain knowledge, and thereby the DA could retrieve some of the knowledge for the data possess.

More information can be found in Fig. 3 about the relationship between the number of remaining patterns and

the number of attributes already integrated. The instances are learnt fastest with descending order of attribute integration, and natural order, random order and ascending order come next. Interestingly, this is not the case for Segment dataset. Although the general trend is similar, the descending order is not the best among four in terms of speed. Therefore, we can conclude that the effect of attribute ordering can be actually problem-dependent.

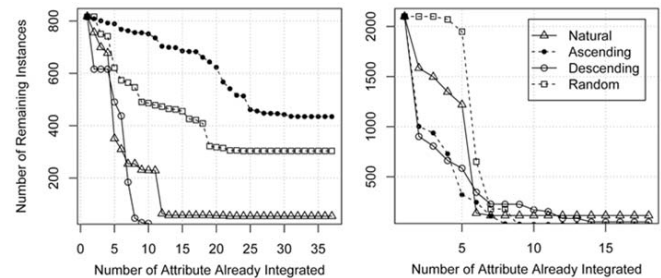


Fig. 3. Effect of attribute integration ordering on anneal & segment

TABLE IV: EFFECT OF ATTRIBUTE INTEGRATION ORDERING ON ANNEAL & SEGMENT

Order	Anneal			Segment		
	CR(SD)	UR	Time	CR(SD)	UR	Time
Natural	90.65	9.24	228.24	61.30	3.40	369.29
Ascending	51.79	47.49	268.19	39.61	4.98	426.92
Descending	97.55	1.98	198.93	87.72	7.34	350.58
Random	75.92	22.13	256.52	81.23	3.04	412.20

## V. DISCUSSION

### A. Strengths of RLGA-IAL

In the previous section, the experimental results were presented with some insightful discussions. The overall performance of RLGA-IAL is shown to be effective in the problems that are of our interests. The results are in accordance with our expectations that the problems of RLGA in higher order dimensional problems can be tackled by integrating it with Incremental Attribute Learning (IAL).

Although the search space of a higher dimensional problem is much larger than the ones with fewer attributes, the performance of RLGA-IAL remains at a consistent level without any parameters fine tuned. This is one major advantage of RLGA-IAL, that it only requires limited memory even with higher dimension problem. This is contrary to the performance of RLGA and Normal GA, as the number of rules in Normal-GA and the population size in RLGA need to be significantly increased accordingly.

Even more impressive is the fact that the RLGA can cope with IAL perfectly. IAL is especially effective with RLGA’s automatic task decomposition feature, as the task can be gradually simplified in the incremental learning process. Therefore, partial solutions are composed by experts with only a subset of attributes. When new attributes are integrated to each expert, it only needs to address a simplified problem. In this manner, we accomplish a feature selection for instances in specific domain. This project also confirms that the RLGA’s potential has not been fully exploited and it could possibly become one of the most powerful rule-based classifiers trained by Genetic Algorithms. Moreover, the

power of IAL is also proved to be effective under different circumstances. The hybrid between input dimension decomposition accomplished by IAL and automatic task decomposition achieved by RLGA leads to the excellent performance of RLGA-IAL.

### B. Existing Weaknesses and Future Development

Although RLGA-IAL has a number of merits, its current status still shows several weaknesses. Efficiency is one of the major issues, as GA-based classifiers generally suffer higher time cost, RLGA-IAL has no improvement in this aspect. The experiments shown in last section actually imply that, although RLGA-IAL results in best classification rate, it also consumes most time. In fact, incrementally training each attribute is inevitably time consuming and the efficiency and accuracy is a tradeoff. Meanwhile, over exploiting each attribute might have a negative effect. The learnt patterns in the training set are generally removed in order to simplify the training task, but it may also prevent instances to be fully utilized in the coming training process.

### C. Future Development

This paper has shown the promising performance of RLGA-IAL. The integration process occurs linearly which means the attribute is integrated one after another. The speed of this process is overall on the order of the attribute number. Suppose we change the integration style from linear to binary, which means the attributes can be aggregated as a binary tree, the speed can be reduced to the logarithmic order. Also, we could integrate multiple attributes each time other than only one of them. This will possibly boost the efficiency as a whole.

The effect of the ordering of integration attributes is still not fully understood. Although the experiments showed the DA is a useful metric, whether it is the best candidate remains to be an open question. Meanwhile, as previously mentioned, current solution on ranking attributes is time-consuming. Therefore, other possible solutions to this problem could be investigated further.

Last but not least, the most difficult part lies in seeking the theoretical foundation of our work. Although we gained some insights, we are still seeking a well-laid theoretical foundation.

## VI. CONCLUSION

To conclude, automatic task decomposition of classification can be implemented in the training process of each attribute through RLGA-IAL. Moreover, IAL could also enhance the searching capability with relatively limited population size. The future work involves determining other possible metrics to order the attributes and also further improve the efficiency of RLGA-IAL.

### ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China under Grant 61070085.

## REFERENCES

- [1] R. Anand, K. Mehrotra, and S. Ranka, "Efficient classification for multiclass problems using modular neural networks," *IEEE Trans. on Neural Networks*, vol. 6, pp. 117-124, January 1995.
- [2] R. Anand, K. Mehrotra, and S. Ranka, "Efficient classification for multiclass problems using modular neural networks," *IEEE Trans. on Neural Networks*, vol. 6, no.1, pp. 117-124, 2002.
- [3] J. H. Ang, S. U. Guan, K. C. Tan, and A. A. Mamun, "Interference-less neural network training," *Neurocomputing*, vol. 71, pp. 3509-3524, October 2008.
- [4] C. Bao, T. N. Neo, and S. U. Guan, "Reduced pattern training in pattern distributor networks," *Journal of Research and Practice in Information Technology*, vol. 39, pp. 273-286, 2007.
- [5] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. 23rd international conference on Machine learning*, 2006.
- [6] A. L. Corcoran and S. Sen, "Using real-valued genetic algorithms to evolve rule sets for classification evolutionary computation," in *Proc. 1st IEEE World Congress on Computational Intelligence*, 1994.
- [7] S. U. Guan, F. Zhu, "Ordered incremental training with genetic algorithms," *International Journal of Intelligent Systems*, vol. 19, pp.1239-1256, 2004.
- [8] L. Fang, S. U. Guan, and H. F. Zhang, "Recursive learning of genetic algorithms with task decomposition and varied rule set," *International Journal of Applied Evolutionary Computation*, vol. 2, no. 4, 2011.
- [9] N. García-Pedrajas, C. Hervás-Martínez and J. Muñoz-Pérez, "Multi-objective cooperative coevolution of artificial neural networks (multi-objective cooperative networks)," *Neural Networks*, vol. 15, no.10, pp.1255-1274, 2002.
- [10] D. E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley, 1989, pp. 41.
- [11] S. U. Guan and S. Li, "Incremental learning with respect to new incoming input attributes," *Neural Processing Letters*, vol. 14, no.3, pp. 241-260, 2001.
- [12] S. U. Guan and J. Liu, "Incremental ordered neural network training," *Journal of Intelligent Systems*, vol. 13, no.1, 2002.
- [13] S. U. Guan and K. Ramanathan, "Percentage-based hybrid pattern training with neural network specific crossover," *Journal of Intelligent Systems*, vol. 16, pp.1-26, 2007.
- [14] S. U. Guan and F. Zhu, "Class decomposition for GA-based classifier agents—a Pitt approach," *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, Cybernetics, vol. 34, no. 1, pp. 381-392, 2004.
- [15] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classier systemsfor multidimensional pattern classification problems," *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Cybernetics, vol. 29, no. 5, pp. 601-618, Oct 1999.
- [16] K. Ramanathan and S. U. Guan, "Recursive pattern based hybrid supervised training," *Engineering Evolutionary Intelligent Systems*, vol.82, pp.129-156, 2008.
- [17] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers—a survey," *IEEE Transactions on Systems, Man, and Cybernetics*, Part C, vol. 35, no 4, pp. 476-487, 2005.
- [18] C. H. Tan, S. U. Guan, and K. Ramanathan, "Recursive hybrid decomposition with reduced pattern training," *International Journal of Hybrid Intelligent Systems*, vol. 6, no. 3, pp. 135-146, 2009.

**Haofan Zhang** received his B.Sc. from the University of Liverpool. He is currently a postgraduate student at University of Waterloo, Canada.

**Sheng-Uei Guan** received his M.Sc. & Ph.D. from the University of North Carolina at Chapel Hill. He is currently a professor in the computer science and software engineering department at Xi'an Jiaotong-Liverpool University (XJTLU). He is also affiliated with Xi'an Jiaotong University as an adjunct faculty staff. Before joining XJTLU, he was a professor and chair in intelligent systems at Brunel University, UK.

**Mengjun Xu** is currently an undergraduate student at Xi'an Jiaotong-Liverpool University.