Multiobjective Fuzzy Genetics-Based Machine Learning based on MOEA/D with its Modifications

Yusuke Nojima1, Koki Arahari1, Shuji Takemura1, and Hisao Ishibuchi2,1

1 Department of Computer Science and Intelligent Systems
Graduate School of Engineering, Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
{nojima@, koki.arahari@ci., shuji.takemura@ci., hisaoi@}cs.osakafu-u.ac.jp

2 Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China

Abstract—Various evolutionary multiobjective optimization (EMO) algorithms have been used in the field of evolutionary fuzzy systems (EFS), because EMO algorithms can easily handle multiple objective functions such as the accuracy maximization and complexity minimization for fuzzy system design. Most EMO algorithms used in EFS are Pareto dominance-based algorithms such as NSGA-II, SPEA2, and PAES. There are a few studies where other types of EMO algorithms are used in EFS. In this paper, we apply a multiobjective evolutionary algorithm based on decomposition called MOEA/D to EFS for fuzzy classifier design. MOEA/D is one of the most well-known decomposition-based EMO algorithms. The key idea is to divide a multiobjective optimization problem into a number of single-objective problems using a set of uniformly distributed weight vectors in a scalarizing function. We propose a new scalarizing function called an accuracy-oriented function (AOF) which is specialized for classifier design. We examine the effects of using AOF in MOEA/D on the search ability of our multiobjective fuzzy genetics-based machine learning (GBML). We also examine the synergy effect of MOEA/D with AOF and parallel distributed implementation of fuzzy GBML on the generalization ability.

Keywords—Fuzzy classifier design, evolutionary fuzzy systems, MOEA/D, accuracy-oriented scalarizing function.

I. INTRODUCTION

Recently, multiobjective evolutionary optimization (EMO) algorithms have actively been utilized for data mining such as classification, clustering, association rule mining [1], [2]. EMO algorithms can easily handle multiple conflicting objectives simultaneously (e.g., accuracy and complexity for classifier design, support and confidence for association rule mining). Although multiobjective evolutionary data mining is a very active research topic, most studies use Pareto dominance-based EMO algorithms. In the field of multiobjective evolutionary fuzzy systems (MoEFS), we can observe the same trend [3]. There are a few studies where other types of EMO algorithms are utilized for MoEFS. For example, in [4], a multiobjective evolutionary algorithm based on decomposition (MOEA/D) [5] is applied to fuzzy genetics-based machine learning (GBML) [6]. MOEA/D is compared with NSGA-II [7] and its variants. In [8], MOEA/D is applied to an iterative rule learning method for fuzzy classifier design. Its performance is examined for imbalanced data sets together with data preprocessing methods.

MOEA/D is one of the most well-known decomposition-based EMO algorithms [5]. High performance of MOEA/D has already been shown in some studies [9], [10]. In MOEA/D, a multiobjective optimization problem is divided into a number of single-objective optimization problems using a set of uniformly distributed weight vectors in a scalarizing function. Various scalarizing functions such as the weighted-sum, the weighted Tchebycheff, and the penalty-based boundary intersection can be used in MOEA/D. The search performance of MOEA/D strongly depends on the selected scalarizing function. It is also possible to design new scalarizing functions in order to cope with problems at hand [11]-[14]. In this paper, we propose a new scalarizing function of MOEA/D specialized for classifier design. It is called an accuracy-oriented function (AOF). We apply it to our multiobjective FGBML [6] and examine the effects on its search ability through computational experiments. We also apply it to our parallel distributed model of multiobjective FGBML [16] and examine its synergy effect with MOEA/D with AOF on the generalization ability.

The rest of this paper is as follows. We briefly explain our multiobjective FGBML and MOEA/D in Section II. Then, we explain AOF and parallel distributed implementation in Section III. In Section IV, we compare among NSGA-II-based FGBML and MOEA/D-based FGBML algorithms using large data sets. Finally, we conclude this paper in Section V.

II. MULTIOBJECTIVE FGBML

A. A Fuzzy Rule-Based Classifier

Let assume an $n$-dimensional $M$-class data set with $m$ training patterns $x_p = (x_{p1}, ..., x_{pn}), p = 1, 2, ..., m$. For the sake of simplicity, we normalize each attribute into the $[0, 1]$ space. A set of fuzzy if then rules is used for constructing a classifier [17], [18]:

Rule $R_q$: If $x_{p1} = A_{q1}$ and $...$ and $x_{pn} = A_{qn}$
then Class $C_q$ with $CF_q$.

where $R_q$ is a rule label for the $q$th rule, $A_p$ is an $i$th antecedent condition. We use one of prespecified triangular membership functions or “don’t care” condition as an antecedent condition for each attribute. The consequent class $C_q$ and rule weight $CF_q$ are deterministically specified based on the compatibility grade of the antecedent part with the training data [17], [18].

B. Multiobjective Fuzzy Genetics-Based Machine Learning

Our FGBML is the hybrid version of the Pittsburgh-style and Michigan-style GBML algorithms [19]. Its multiobjective
version is also proposed in [6]. The non-dominated ranking and crowding distance of NSGA-II [7] are used for the parent selection and generation update in order to find a number of non-dominated solutions widely spread along the Pareto front. The following bi-objective formulation is used in [6]:

\[ f_1(S): \text{Error rate by a classifier } S \text{ on all training patterns}, \]
\[ f_2(S): \text{Number of rules in the classifier } S, \]

Minimize \( f_1(S) \) and minimize \( f_2(S) \),

where \( S \) is a solution which represents a classifier in our multiobjective FGBML. To calculate the error rates \( f_1(S) \) on the training patterns, we employ a winner-take-all method.

The three-objective formulation is also used by adding an additional complexity measure:

\[ f_3(S): \text{Total rule length of the classifier } S, \]

Minimize \( f_1(S) \), minimize \( f_2(S) \), and minimize \( f_3(S) \),

where the total rule length is calculated by the number of fuzzy membership functions in the classifier.

C. MOEA/D

There are three main characteristics of MOEA/D [5]. One is to handle a multiobjective problem as a collection of a number of single-objective problems. Each single-objective problem has a scalarizing function with a different weight vector. An individual is assigned to each weight vector. Thus the population size is the same as the number of weight vectors. A set of weight vectors are uniformly distributed in order to maintain the diversity of the population like Fig. 1.

As a scalarizing function, the weighted Tchebycheff function is often used in the literature.

\[ \min \left\{ \lambda^T \cdot \left( z^* - f_i(x) \right) \right\}, \quad i = 1, 2, \ldots, N, \]

where \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N) \) and \( z^* = (z^*_1, z^*_2, \ldots, z^*_N) \) are the number of objectives, the weight vector, and a reference point in the objective space, respectively. In the case of classifier design, we specify the reference point as the ideal point (i.e., the minimum value for each objective function) found so far.

Another characteristic is local selection. Neighbors are defined for each weight vector by the distance between weight vectors. In the case of Fig. 1, when the neighborhood size is specified as five, the neighbors of the red vector (i.e., the fifth vector from left) are four blue vectors (i.e., the third, fourth, sixth and seventh vectors from left). The own vector (i.e. the red vector) is also regarded as a neighbor. In the local selection, two parent individuals are selected from the neighbor vectors.

The other characteristic is local competition. A newly generated individual is compared with each of its neighbor individuals. If the newly generated individual is better with respect to the neighbor’s scalarizing function, the neighbor individual is replaced with the newly-generated one. It may occur that all neighbor individuals are replaced with the single new individual.

III. MOEA/D WITH ITS MODIFICATIONS

A. Accuracy-Oriented Scalarizing Function

In the literature, the weighted Tchebycheff function is frequently used in various studies. Although the Tchebycheff function has nice theoretical property and shows its strength, there is a problem for classifier design. As shown in Fig. 2, a small number of weight vectors correspond to better individuals in terms of \( f_3(S) \) (i.e., accurate classifiers). Thus, it is difficult to maintain individuals around the low error region (i.e., the red circle in Fig. 2).

In order to bias the search direction toward the low error region, we propose an accuracy-oriented function (AOF) for classifier design. First, we define the weight vectors like Fig. 3 (a). We use multiple reference points (e.g., for the two-objective formulation, \( z^*_1 = (0, 1), z^*_2 = (0, 2), \ldots, z^*_\text{max} = (0, N_{\text{max}}) \)). \( N_{\text{max}} \) is the maximum number of rules in a classifier. From the reference points, we define the weight vectors \( \lambda^1 = \lambda^2 = \ldots = \lambda^{\text{max}} = (1, 0) \). For the \( k \)th vector, AOF is defined as follows:

\[ \min g_k^{\text{AOF}}(x) = f_1(x) + P + f_2(x) / m, \]

where \( m \) is the number of patterns. \( P \) is the penalty term:

\[ P = 100 \max \left\{ f_2(x) - k, 0 \right\}. \]

The second term of (5) plays a role in finding more accurate classifiers with \( k \) rules or less. For example, the fifth vector from left (i.e., \( k = 5 \)) tends to have an individual which represents a classifier with five rules. Using AOF, we expect
that it is easy to maintain individuals around the low error region like Fig. 3 (b).

The coefficient of the second term in (5) and (6) should be carefully specified. More appropriate values and/or automatic specification are left for an extended study.

In [23], we applied our parallel distributed implementation to NSGA-II based multiobjective FGBML [6]. The quadratic speed up can be achieved. However, the rotation of training data subsets strongly affected the number of rules in the accurate and complicated classifiers.

IV. COMPUTATIONAL EXPERIMENTS

A. Experimental Setting

To examine the effects of using AOF in MOEA/D and parallel distributed implementation on the search ability of our multiobjective FGBML algorithms, we used four large data sets available from Keel data repository [20], [21] in Table I.

<table>
<thead>
<tr>
<th>Data</th>
<th>No. patterns</th>
<th>No. attributes</th>
<th>No. classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phoneme</td>
<td>5404</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Ring</td>
<td>7400</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>Satimage</td>
<td>6435</td>
<td>36</td>
<td>6</td>
</tr>
<tr>
<td>Segment</td>
<td>2079</td>
<td>18</td>
<td>7</td>
</tr>
</tbody>
</table>

The common parameter settings of NSGA-II and MOEA/D for non-parallel and non-distributed models are as follows:

Population size: 180,
Crossover: Uniform crossover,
Crossover probability: 0.9 (Pittsburgh), 0.9 (Michigan),
Mutation probability: \(\frac{1}{n|S|}\) (Pittsburgh), \(\frac{1}{n}\) (Michigan),
\(n\): Number of attributes,
Michigan operation probability: 0.5,
Number of rules: 30 (Initial), 1 (Minimum), 60 (Maximum),
Termination condition: 50,000 generations,
Number of runs: 50 (i.e., 5 x 10-fold cross validation).

For fair comparison with NSGA-II, the population size was specified as 180 for EMO algorithms. In the case of MOEA/D with AOF, three weight vectors were allocated from each reference point. In addition, the neighborhood size of MOEA/D was specified as 10% of each subpopulation size.

For parallel distributed models, we used three CPUs as shown in Fig. 4. The expected speed up was about nine times faster than the non-parallel non-distributed models. The rotation of training data subsets and individual migration were performed every 100 generations.

As antecedent conditions of fuzzy if-then rules, we used homogeneous triangular membership functions with four different granularities and a \(\text{don’t care}\) condition in Fig. 5.
B. Effects of Using AOF in MOEA/D

Figs. 6-8 show the comparison of NSGA-II, MOEA/D with the weighted Tchebycheff function (MOEA/D-Tch), and MOEA/D with AOF (MOEA/D-AOF) for the four data sets. All three methods are non-parallel non-distributed models. Each symbol represents the average error rate over the non-dominated classifiers with the corresponding number of fuzzy rules which were obtained from more than 25 times among the 50 runs. Closed inverted triangles represent the error rates of the most complicated classifiers. Each color corresponds to each method (i.e., Grey, blue, and green represent NSGA-II, MOEA/D-AOF, and MOEA/D-Tch, respectively).

For the training data set of the Phoneme data, we can observe the strong search ability of NSGA-II comparing with MOEA/D-Tch in Fig. 6. We can also observe that MOEA/D-AOF improved the search ability of MOEA/D. For the test data set, the classifiers obtained by NSGA-II were slightly more accurate than the others.

For the Ring data, the same trend as the Phoneme data was observed for the training data set in Fig. 7 (a). However, the error rate on the test data by MOEA/D-Tch was the best among three methods in Fig. 7 (b).

For the Satimage data and the Segment data in Figs. 8 and 9, more accurate classifiers were obtained by MOEA/D-AOF for the training data. For the test data, most of classifiers obtained by MOEA/D-Tch dominated those by the others.

C. Effects of Parallel Distributed Implementation

We examined the synergy effects of MOEA/D-AOF and our parallel distributed implementation on the generalization ability of multiobjective FGBML. Figs. 10-13 show the comparison of three parallel distributed models.

Comparing with the standard models in Figs. 6-9, complicated classifiers with a large number of rules were not obtained by the parallel distributed models. On the other hand, the overfitting phenomena were not observed from Figs. 10-13.

For the Phoneme data, we can observe the improvement of the search performance on the training data by MOEA/D-AOF. The generalization ability of MOEA/D-AOF on the test data was also improved.
For the Ring data in Fig. 11, MOEA/D-Tch was the best method among three methods. Simpler and more accurate classifiers were obtained by MOEA/D-Tch for both the training data and the test data.

For the Satimage data and the Segment data in Figs. 12 and 13, we can observe that simpler and more accurate classifiers were obtained by MOEA/D-AOF for both the training data and the test data.

Table II shows the error rates on the test data sets of the most accurate classifiers with respect to the training data sets (i.e., the most complicated classifiers with the largest number of rules). The values are corresponding to the closed inverted triangles in Figs. 6-13. We performed Wilcoxon signed rank tests ($\alpha = 0.05$) between the standard model and the parallel distributed model. The * mark in Table II represents that there was no statistical difference. From Table II, we can clearly see that only the parallel distributed version of MOEA/D-AOF maintained the generalization ability on the test data set for all four data (i.e., no statistical difference between the standard model and the parallel distributed model for all data sets). Table III shows that the most accurate classifiers with respect to the training data sets obtained by the parallel distributed models had a smaller number of rules than the standard models.

V. CONCLUSIONS

In this paper, we proposed an accuracy-oriented function (AOF) as a new scalarizing function of MOEA/D. We applied MOEA/D with AOF to our multiobjective FGBML for fuzzy classifier design. From the experimental results using four large data sets, we observed a clear improvement of the search ability of MOEA/D by using AOF for the training data. We also applied MOEA/D with AOF to our parallel distributed models of FGBML in order to examine their positive synergy. From the experimental results, we observed good synergy effects of MOEA/D with AOF and the parallel distributed implementation for all data sets. The parallel distributed version of multiobjective FGBML based on MOEA/D with AOF obtained simpler classifiers (i.e., a smaller number of rules in a classifier) for much shorter computation time while maintaining the test data accuracy.

It is obvious that there was no best EMO algorithm for any data sets. This implies the necessity of the algorithm selection or the scalarizing function selection for each data. An appropriate algorithm and scalarizing function may strongly depend on the tradeoff curve between the objective functions (e.g., error and the number of rules). A repeated double cross validation [22] may also be useful for this task. Use of multiple scalarizing functions in MOEA/D would also be a promising approach. These are left for future research topics. For parallel distributed implementation of EMO algorithms, NSGA-II with cone separation [24] and MOEA/D-M2M [25] could be alternative methods.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers 25330292 and 16K00336.

REFERENCES


<table>
<thead>
<tr>
<th>TABLE II. THE ERROR RATES ON TEST DATA SETS OF THE MOST ACCURATE CLASSIFIERS WITH RESPECT TO TRAINING DATA SETS.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phoneme</strong></td>
</tr>
<tr>
<td>NSGA-II</td>
</tr>
<tr>
<td>Standard</td>
</tr>
<tr>
<td>14.44</td>
</tr>
<tr>
<td>4.26*</td>
</tr>
<tr>
<td>13.38*</td>
</tr>
<tr>
<td>5.14</td>
</tr>
<tr>
<td>* represents no statistical difference between the standard model and the parallel distributed model.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III. THE NUMBER OF RULES IN THE MOST ACCURATE CLASSIFIERS WITH RESPECT TO THE TRAINING DATA SETS.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phoneme</strong></td>
</tr>
<tr>
<td>NSGA-II</td>
</tr>
<tr>
<td>Standard</td>
</tr>
<tr>
<td>54.30</td>
</tr>
<tr>
<td>53.52</td>
</tr>
<tr>
<td>46.36</td>
</tr>
<tr>
<td>25.96</td>
</tr>
</tbody>
</table>

**MOEA/D**: Multiobjective Evolutionary Algorithm
**NSGA-II**: Non-dominated Sorting Genetic Algorithm II
**Tchebycheff**: Tchebycheff approach
**GBML**: Genetic-Based Multi-Layered

**Knowledge Extraction based Evolutionary Learning (KEEL)**: A software tool for data mining and evolutionary learning.