Multiobjective Data Mining from Solutions by Evolutionary Multiobjective Optimization

Yusuke Nojima, Yuki Tanigaki, and Hisao Ishibuchi

1INTRODUCTION

In general, real-world problems have multiple objectives to be simultaneously optimized. Because they are often conflicting with each other, evolutionary multiobjective optimization (EMO) algorithms have frequently been used to find a number of nondominated solutions which approximate the Pareto front [6]. Then, a decision maker chooses one of the non-dominated solutions according to her/his preference. Another usage of the obtained non-dominated solutions is to analyze the relationship between the design variables and the objective functions for the optimization problems [4], [7], [9]. For this purpose, data mining techniques have been used to extract the knowledge from the solution set obtained by EMO algorithms as a post-analytical process in the literature (e.g., self-organizing map [16], ANOVA [13], heatmap [17], rough sets [18], clustering [3], [19], rule mining [2], learning classifier systems [15]).

From a practical point of view, this process itself should be considered as a multiobjective problem because there is a tradeoff relationship between the accuracy and complexity of knowledge. Highly accurate knowledge is usually complicated, while simple knowledge is less accurate. Moreover, an appropriate tradeoff strongly depends on its user and cannot be specified beforehand.

In this paper, we propose a new framework of multiobjective data mining from solutions evaluated by an EMO algorithm. The proposed framework is based on the selection of multiple target regions of interest and multiobjective genetic fuzzy rule selection [1], [11], [12]. Although only non-dominated solutions have frequently been used for the post-analytical process in most of the previous studies, we use all the solutions evaluated during the execution of an EMO algorithm. Multiple regions of interest are specified by a user in the objective space. Each region with a number of solutions is handled as a different class. Then, the combination of fuzzy if-then rules is optimized by another EMO algorithm in order to generate a number of non-dominated classifiers with respect to accuracy and complexity.

This paper is organized as follows. In Section 2, we explain the proposed framework of multiobjective data mining for the post-analytical process. In Section 3, we apply the proposed framework to two engineering problems. Finally, this paper is concluded in Section 4.
2 MULTIOBJECTIVE DATA MINING

Figure 1 shows the outline of the proposed multiobjective data mining framework. The proposed framework can be divided into three stages: (1) specification of regions of interest, (2) classification rule mining, and (3) multiobjective classifier design. Through these three stages, users can obtain multiple classifiers as knowledge which represents the relationship between design variables and objective functions for optimization problems. The following subsections explain each stage in detail.

2.1 Specification of Regions of Interest

By applying an EMO algorithm to a multiobjective problem, a number of non-dominated solutions are obtained together with dominated solutions. We use all the evaluated solutions in the execution of the EMO algorithm for data mining because we assume that the dominated solutions also have important information in order to analyze the relationship between the design variables and objective functions. The use of every solution would be necessary for some problems where each solution evaluation is computationally expensive.

At the first stage of the proposed framework, a user first specifies multiple regions of interest in the objective space. Then, each region with a number of solutions is handled as a different class (i.e., the number of classes is the same as the number of the regions). A set of solutions with class labels is regarded as a dataset in a classification problem where each solution in the decision space is handled as a pattern. For simplicity, the domain of each design variable is normalized to [0, 1] based on the minimum and maximum values in the data.

Figure 1 shows an example if the user wants to know the difference between two extreme regions; solutions with low $f_1$ value (Class 1) and solutions with low $f_2$ value (Class 2). If the user wants to know the difference between the Pareto optimal solutions and others, we can specify two regions: (near-) Pareto optimal solutions as Class 1 and non-Pareto optimal solutions as Class 2 shown in Fig. 2 (a). If the user wants to analyze the characteristics of solutions at different generations, we can specify three regions: an initial population as Class 1, a middle population as Class 2, and a final population as Class 3 shown in Fig. 2 (b).

2.2 Classification Rule Mining

Multiobjective genetic fuzzy rule selection [12] can be divided into two phases: classification rule mining and multiobjective classifier design. These two phases are corresponding to the second and third stages in the proposed framework.

In classification rule mining, the following fuzzy if-then rules are extracted from the dataset generated in Subsection 2.1.

If $x_1$ is $A_{q_1}$ and ... and $x_n$ is $A_{q_n}$, then Class $C_q$ with $C_{F_q}$.

where $q$ is the rule index, $x_i$ ($i = 1, 2, ..., n$) is the normalized value of the $i$-th design variable, $n$ is the number of the design variables, $A_{q_i}$ is the antecedent fuzzy set for the $i$-th design variable of the $q$-th rule. $C_q$ and $C_{F_q}$ are a consequent class label and a rule weight of the $q$-th rule, respectively. For the antecedent fuzzy sets, we use triangular membership functions with four different granularities (i.e., 14 membership functions in total) shown in Fig. 3. In addition, “don’t care” is also handled as a special fuzzy set which is the same as the interval [0, 1]. One out of 15 antecedent fuzzy sets is assigned to each design variable in a rule. Thus, the maximum number of combinations for a single rule is $15^n$. The “don’t care” is useful for generating generalized rules.

The consequence part $C_q$ and $C_{F_q}$ can be determined according to the compatibility grade of the antecedent part $A_q = (A_{q_1}, A_{q_2}, ..., A_{q_n})$ with training data. The compatibility grade $\mu_{A_q}(x_p)$ for the pattern $x_p = (x_{p_1}, x_{p_2}, ..., x_{p_m})$ is calculated by the product operator:

$$\mu_{A_q}(x_p) = \mu_{A_{q_1}}(x_{p_1}) \cdot \ldots \cdot \mu_{A_{q_n}}(x_{p_n}).$$

The “confidence” and “support” are often used in association rule mining. For the $h$-th class, the confidence $c(A_q \Rightarrow \text{Class } h)$ and support $s(A_q \Rightarrow \text{Class } h)$ are defined as [11]:

![Figure 1: The outline of the proposed framework.](image-url)
the largest confidence value.

A classifier is represented by a binary string where minority class are less than 300. Thus, less than 300 rules per class with respect to the product of the confidence level. From the rules which meet these levels, we select the best.

We set the minimum confidence level and the minimum support values. In some cases, the selected rules for the minority class are less than 300. Thus, less than 300 rules per class with respect to the product of the confidence level. From the rules which meet these levels, we select the best.

The common parameters of multiobjective genetic fuzzy rule selection for both problems are as follows:

**Classification rule mining**
- Maximum rule length: 3
- Minimum confidence level: 0.5
- Minimum support level: 0.02
- Maximum number of extracted rules: 300 per class.

**Multiobjective classifier design**
- Optimizer: NSGA-II
- Population size: 200
- Crossover: Uniform crossover (Probability: 0.9)
- Mutation: Bit-flip mutation (Probability: $1/n$, $n$: Gene length), Number of generations: 2,000.

Most parameters are specified according to the referenced studies. Appropriate parameters may exist for each problem. For a demonstration purpose, we leave this issue for a future study.

**3.2 Welded Beam Design Problem**

The welded beam design problem has four design variables, two objective functions, and four constraints [9]. In this problem, two beams are welded to carry a certain load $F$ in Fig. 4. The design variables are the thickness of the beam $b$, the width of the beam $t$, the length of weld $l$, and the weld thickness $h$. Thus, $x = (b, l, t, h)$. The objective functions are the cost of the beam $f_1(x)$ and the vertical deflection at the end of the beam $f_2(x)$ as follows:
Minimize $f_1(x) = 1.10471h^2l + 0.04811lb(14.0 + l)$, \hspace{1cm} (9)

Minimize $f_2(x) = \frac{2.1952}{l^3b}$, \hspace{1cm} (10)

$0.125 \leq h, b \leq 5.0$, \hspace{1cm} (11)

$0.1 \leq l, t \leq 10.0$. \hspace{1cm} (12)

The constraints and other detailed information can be found in [9].

Figure 4: The welded beam design problem.

We utilized jMetal (Version 4.5, http://jmetal.sourceforge.net) to implement and solve this problem by NSGA-II. The parameters of NSGA-II were as follows:

- Population size: 200,
- Number of fitness evaluations: 100,000,
- Crossover: SBX (Probability: 1.0),
- Mutation: Polynomial mutation (Probability: 0.25).

All the solutions evaluated by NSGA-II were archived. The replicated solutions were removed from the archive. Figure 5 shows the solutions in the objective space. The minimum and maximum values of each design variable are summarized in Table 1. All the values of the design variables were normalized in [0, 1] using the values in Table 1.

Table 1: The range of design variables of solutions obtained by NSGA-II for the welded beam design problem.

<table>
<thead>
<tr>
<th>Design variable</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>0.171</td>
<td>4.541</td>
</tr>
<tr>
<td>$l$</td>
<td>0.429</td>
<td>9.972</td>
</tr>
<tr>
<td>$t$</td>
<td>2.171</td>
<td>10.000</td>
</tr>
<tr>
<td>$b$</td>
<td>0.238</td>
<td>5.000</td>
</tr>
</tbody>
</table>

We examined two cases for different class specifications for the welded beam design problem.

**Class Specification 1**

First, we assumed that a user wants to know the difference between solutions in two extreme regions, the best regions on $f_1(x)$ and $f_2(x)$. According to this assumption, we assigned class labels to the solutions around those two regions. They are highlighted by red and blue in Fig. 6. There were 399 patterns in Class 1 and 1,983 patterns in Class 2.

Figure 5: Solutions evaluated by NSGA-II for the welded beam design problem.

Figure 6: Class specification 1 (two extreme regions) for the welded beam design problem.

Figure 7 shows the non-dominated classifiers obtained by multiobjective classifier design for the class specification 1.

Figure 7: The classifiers obtained by multiobjective classifier design for the class specification 1.

If $b$ is Small$^1$ then Class 1 with 1.00,
If $b$ is Large$^2$ then Class 2 with 0.99.
“Small” and “Large” are the membership functions in Fig. 3. The superscript represents the granularity (i.e., the number of partitions). This classifier correctly classified every pattern. It is obvious that the cost and the deflection strongly depend on the thickness of the beam \( b \). If the thickness is small, the cost is minimized. If the thickness is large, the deflection is small.

Class Specification 2

Next, we assumed that a user wants to know the difference between the Pareto optimal solutions and non-Pareto optimal solutions. According to this assumption, we assigned Class 1 to the Pareto optimal solutions and Class 2 to the non-Pareto optimal solutions. Class 1 and Class 2 are highlighted by red and blue in Fig. 8. The number of Class 1 patterns was 1,706, while the number of Class 2 patterns was 430.

Figure 8: Class specification 2 (Pareto or non-Pareto) for the welded beam design problem.

Figure 9 shows the non-dominated classifiers obtained by multiobjective classifier design. The simplest classifier has only two conditions and two rules as follows:

- If \( h \) is Very Small then Class 1 with 0.71,
- If \( t \) is Medium then Class 2 with 1.00.

The accuracy of this classifier was 96.40%. The confusion matrix is shown in Table 2. Most of the Pareto optimal solutions can be correctly classified by this classifier.

The most accurate classifier has eight conditions and six rules as follows:

- If \( l \) is Small and \( b \) is Large then Class 1 with 0.74,
- If \( t \) is Small then Class 2 with 1.00,
- If \( h \) is Large then Class 2 with 0.99,
- If \( l \) is Large then Class 2 with 0.99,
- If \( t \) is Medium and \( b \) is Medium then Class 2 with 0.42,
- If \( t \) is Small then Class 2 with 1.00.

The accuracy of this classifier was 99.67%. The confusion matrix is shown in Table 3. The above examples clearly show that our approach can obtain linguistically interpretable simple and highly accurate classifiers for the welded beam design problem.

3.3 Conceptual Design Optimization of Hybrid Rocket Engine

The hybrid rocket engine uses a propellant stored in two kinds of phases, liquid oxidizer and solid fuel [14]. It has both advantages of the liquid and solid rockets. In this paper, we used the solutions evaluated by MOGA [10] available from:


The conceptual design optimization problem of hybrid rocket engine has mainly six design variables: the mass flow of oxidizer \( m_{\text{oxi}} \), the fuel length \( L_{\text{fuel}} \), the port radius of fuel \( r_{\text{port}} \), the combustion time \( t_{\text{burn}} \), the pressure of combustion chamber \( P_c \), and the aperture ratio of nozzle \( \varepsilon \). There are two main objective functions: the flight altitude \( H_{\text{max}} \) and \( f_1(x) \) and the total gross weight \( M_{\text{tot}} \).

Figure 9: The classifiers obtained by multiobjective classifier design for the class specification 2.

The minimum and maximum values of each design variable are summarized in Table 4. All the values of the design variables were normalized in \([0, 1]\).

We examined two cases for different class specifications for this problem as well as the previous problem.

Class Specification 3

For this problem, we assumed that a user wants to know the difference among solutions in three extreme regions, the best
region on \( f_1(x) \), the best region on \( f_2(x) \), and the worse region on both objectives. According to this assumption, we assigned Class 1, Class 2, and Class 3 to solutions around the abovementioned three regions as highlighted respectively by red, blue and green, and shown in Fig. 11. The number of patterns for each class was as follows: 88 (Class 1), 91 (Class 2), and 129 (Class 3).

Figure 10: Solutions evaluated by MOGA for the concept design optimization of hybrid rocket engine.

Table 4: The range of design variables of solutions obtained by MOGA for the conceptual design optimization of hybrid rocket engine.

<table>
<thead>
<tr>
<th>Design variable</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{\text{oxi}} )</td>
<td>1.017</td>
<td>30.000</td>
</tr>
<tr>
<td>( L_{\text{fuel}} )</td>
<td>1.008</td>
<td>9.998</td>
</tr>
<tr>
<td>( r_{\text{port}} )</td>
<td>10.056</td>
<td>199.718</td>
</tr>
<tr>
<td>( t_{\text{burn}} )</td>
<td>15.000</td>
<td>34.989</td>
</tr>
<tr>
<td>( P_{c} )</td>
<td>30.010</td>
<td>39.996</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>5.000</td>
<td>6.999</td>
</tr>
</tbody>
</table>

Figure 11: Class specification 3 (three extreme regions) for the conceptual design optimization of hybrid rocket engine.

Figure 12 shows the non-dominated classifiers obtained by multiobjective classifier design for the class specification 3.

Class Specification 4

Finally, we assumed that a user wants to know the difference between the (near-) Pareto optimal solutions and other solutions for the conceptual design optimization of hybrid rocket engine. According to this assumption, we assigned Class 1 to the (near-) Pareto optimal solutions as highlighted by red in Fig. 13. We also assigned Class 2 to other solutions as highlighted by blue in Fig. 13. The number of Class 1 patterns was 170, while the number of Class 2 patterns was 844.

Figure 14 shows the non-dominated classifiers obtained by multiobjective classifier design. The simplest classifier has only two conditions and two rules as follows:

- If \( m_{\text{oxi}} \) is Very Small then Class 1 with 0.57,
- If \( L_{\text{fuel}} \) is Large and \( r_{\text{port}} \) is Small then Class 1 with 0.55,
- If \( L_{\text{fuel}} \) is Large and \( t_{\text{burn}} \) is Large then Class 2 with 0.91.

From this result, we can say that the three-class data (i.e., regions of interest) can be characterized by only three if-then rules.
The accuracy was 97.5%. The number of rules is three. Table 6 shows the confusion matrix. At the small risk of interpretability loss (i.e., an increase in the complexity), the accuracy was clearly improved by adding only one rule and three conditions.

Figure 13: Class specification 4 (Pareto or non-Pareto) for the conceptual design optimization of hybrid rocket engine.

Figure 14: The classifiers obtained by multiobjective classifier design for the class specification 4.

Table 5: The confusion matrix of the classifier with length 2.

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Rejection</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>107</td>
<td>63</td>
<td>0</td>
</tr>
<tr>
<td>Class</td>
<td>1</td>
<td>842</td>
<td>1</td>
</tr>
</tbody>
</table>

The feedback from the engineers in companies: Usually, rules with one condition are too simple and obvious. On the other hand, rules with many conditions are not understandable. Thus, rules with two or three conditions are interesting as knowledge.

3.4 Sensitivity of Class Specifications

The difficulty of classifier optimization problems depends on how users specify multiple regions of interest. We additionally examined a new class specification 5 which is similar to the class specification 2. We assigned Class 2 to more solutions near Pareto-optimal solutions shown in Fig. 15. The number of Class 2 patterns increased from 430 to 568. Figure 16 shows the non-dominated classifiers obtained by multiobjective classifier design. The circles are the same as Fig. 9. The triangles represent classifiers for the class specification 5. We can observe different tradeoffs between the accuracy and complexity of classifiers for two class specifications, even though the distribution of Class 2 patterns in the class specification 5 was very similar to that in the class specification 2.

The simplest classifier has only two conditions and two rules as follows:

If $m_{oxi}$ is Very Small\textsuperscript{4} then Class 1 with 0.61,
If $h$ is Medium\textsuperscript{4} then Class 2 with 1.00.

The accuracy of this classifier was 92.39%. Interestingly, the antecedent conditions and class labels of the above rules were the same as those in the simplest classifier obtained in the class specification 2, although the rule weights were different.

On the other hand, the most accurate classifier has 14 conditions and eight rules as follows:

If $l$ is Small\textsuperscript{1} and $b$ is Large\textsuperscript{4} then Class 1 with 0.57,
If $m_{oxi}$ is Very Small\textsuperscript{1} then Class 1 with 0.74,
If $L_{fuel}$ is Large\textsuperscript{4} and $t_{burn}$ is Large\textsuperscript{1} then Class 1 with 0.91,
If $L_{fuel}$ is Small\textsuperscript{5} and $r_{por}$ is Medium\textsuperscript{1} then Class 1 with 0.55,
If $m_{oxi}$ is Very Large\textsuperscript{5} then Class 2 with 0.98,
If $m_{oxi}$ is Large\textsuperscript{3} and $\varepsilon$ is Small\textsuperscript{2} then Class 2 with 0.97,
If $L_{fuel}$ is Medium\textsuperscript{1} and $r_{por}$ is Small\textsuperscript{5} and $t_{burn}$ is Large\textsuperscript{2} then Class 2 with 0.82,
If $m_{oxi}$ is Large\textsuperscript{4} and $t_{burn}$ is Large\textsuperscript{3} then Class 2 with 0.97.

Table 7: The confusion matrix of the most accurate classifier.

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Rejection</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>159</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>Class</td>
<td>2</td>
<td>844</td>
<td>0</td>
</tr>
</tbody>
</table>
If \( i \) is Medium\(^2 \) and \( b \) is Large\(^4 \) then Class 2 with 0.70.
If \( i \) is Medium\(^7 \) and \( b \) is Large\(^3 \) then Class 2 with 0.77.

The accuracy of this classifier was 97.80\%. Five rules highlighted by bold face were commonly used in the most accurate classifier in the class specification 2.

From this result, apart from the classification accuracy, we can provide common information on the relationship between the design variables and objective functions to users.

![Figure 15: Class specification 5 (Pareto or non-Pareto) for the welded beam design problem.](image)

**Figure 15**: Class specification 5 (Pareto or non-Pareto) for the welded beam design problem.

<table>
<thead>
<tr>
<th>Accuracy (%)</th>
<th>( F_1(S) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>2</td>
</tr>
<tr>
<td>92</td>
<td>4</td>
</tr>
<tr>
<td>94</td>
<td>6</td>
</tr>
<tr>
<td>96</td>
<td>8</td>
</tr>
<tr>
<td>98</td>
<td>12</td>
</tr>
<tr>
<td>100</td>
<td>14</td>
</tr>
</tbody>
</table>

![Figure 16: The classifiers obtained by multiobjective classifier design for the class specification 5.](image)

**Figure 16**: The classifiers obtained by multiobjective classifier design for the class specification 5.

4 CONCLUSIONS

We proposed a new simple framework for the post-analytical process for understanding the relationship between the design variables and objective functions of solutions evaluated by EMO algorithms. We demonstrated the characteristics of the proposed framework using two engineering problems. There are two main features. One is that users can freely specify multiple regions of interest. The other is that a number of non-dominated classifiers with a different tradeoff between accuracy and complexity can be obtained and provided to the users.

Although we used three objective functions related to the accuracy and complexity in multiobjective classifier design, we can also use other measures as objective functions. For example, if the data is class imbalance, the specialized measures such as F-score, Kappa, and AUC should be used [5]. The interpretability of the knowledge should also be discussed more from the practical point of view. Using various types of antecedent sets like intervals, rough sets, Type-II fuzzy sets would also be interesting to study in the future.

The number of design variables for the problems used in this paper is very small (i.e., four for the first problem, and five for the second problem). The scalability of the proposed framework for high-dimensional problems would be examined as future work.

REFERENCES