Y. Nojima, Y. Tanigaki, and H. Ishibuchi, "Multiobjective data mining from solutions by evolutionary multiobjective optimization," Proc. of 2017 Genetic and Evolutionary Computation Conference, pp. 617-624, Berlin, Germany, July 15-19, 2017.

Multiobjective Data Mining from Solutions by Evolutionary Multiobjective Optimization

Yusuke Nojima[1)], Yuki Tanigaki[1)]
[1)] Osaka Prefecture University
Sakai, Osaka, Japan
{nojima@, yuki.tanigaki@ci., hisaoi@}cs.osakafu-u.ac.jp

Hisao Ishibuchi[2) 1)]
[2)] Southern University of Science and Technology (SUSTech)
Shenzhen, Guangdong, China

ABSTRACT

One research direction in the field of evolutionary multiobjective optimization (EMO) is a post-analytical process of non-dominated solutions in order to analyze the relationship between design variables and objective functions for optimization problems. For this purpose, data mining techniques have been used in some studies. From a practical point of view, this process itself should be considered as a multiobjective optimization problem. In this paper, multiobjective genetic fuzzy rule selection is applied to the post-analytical process of solutions obtained by EMO algorithms. First, multiple regions of interest are specified in the objective space. Each region with a number of solutions is handled as a different class. A set of patterns is generated by the labeled solutions. Second, a number of fuzzy if-then rules are generated by classification rule mining. Finally, an EMO algorithm is applied to combinatorial optimization of fuzzy if-then rules in order to obtain a number of non-dominated fuzzy classifiers with respect to accuracy and complexity. Through computational experiments using two engineering problems, we show that we can obtain various classifiers with a variety of complexity-accuracy tradeoff.

CCS CONCEPTS

• **Mathematics of computing → Optimization algorithms**

KEYWORDS

Evolutionary multiobjective optimization, data mining, genetic fuzzy rule selection, pattern classification.

ACM Reference format:

Y. Nojima, Y. Tanigaki, and H. Ishibuchi. 2017. Multiobjective Data Mining from Solutions by Evolutionary Multiobjective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017, Berlin, Germany, July 15–19, 2017 (GECCO '17), 8 pages.*
DOI: 10.1145/3071178.3080293

ACM ISBN 978-1-4503-4920-8/17/07...$15.00
http://dx.doi.org/10.1145/3071178.3080293

1 INTRODUCTION

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 non-dominated 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 dataset for data mining is generated by the labeled solutions. In multiobjective genetic fuzzy rule selection, first a pre-specified number of fuzzy if-then rules are generated by classification rule mining from the dataset. 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.

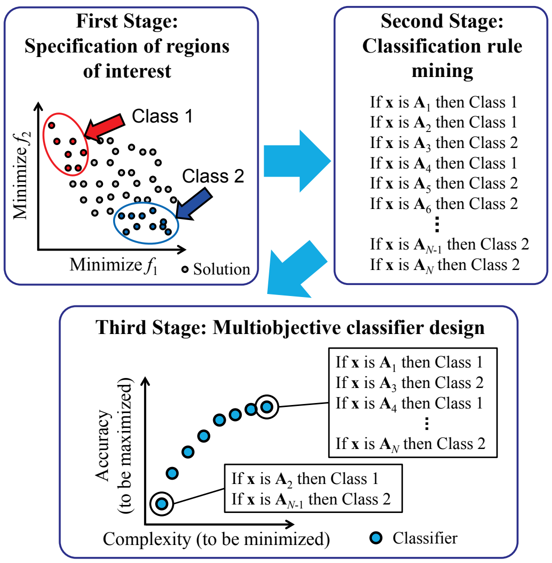

Figure 1: The outline of the proposed framework.

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).

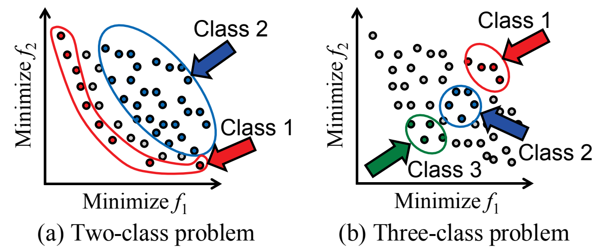

Figure 2: Examples of the specification of regions of interest.

For two-objective problems, multiple regions of interest can be easily chosen like Figs. 1 and 2. For three- and many-objective problems, it is possible to choose them using simple conditions. An example is to choose (near-) Pareto optimal solutions (Class 1) and others (Class 2). Another example is to choose the best 20 solutions for each objective function (Class 1, Class 2, ..., Class L; L: the number of objective functions).

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.

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

where 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_{qi} is the antecedent fuzzy set for the i-th design variable of the q-th rule. C_q and CF_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 CF_q can be determined according to the compatibility grade of the antecedent part $\mathbf{A}_q = (A_{q1}, A_{q2}, \ldots, A_{qn})$ with training data. The compatibility grade $\mu_{\mathbf{A}_q}(\mathbf{x}_p)$ for the pattern $\mathbf{x}_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ is calculated by the product operator:

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \ \ldots\ \cdot \mu_{A_{qn}}(x_{pn})\,. \tag{2}$$

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

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{\sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{\sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}, \tag{3}$$

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{\sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{m}, \tag{4}$$

where m is the number of training patterns.

The consequent class C_q of the q-th rule is the class which has the largest confidence value.

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max_{h=1,2,\ldots,M} \{c(\mathbf{A}_q \Rightarrow \text{Class } h)\}. \tag{5}$$

The rule weight CF_q is defined as:

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h). \tag{6}$$

$CF_q = 1$ means the q-th rule correctly covers only patterns with Class C_q. If $CF_q \leq 0$, the q-th rule is not generated.

Although 15^n rules can be generated in the heuristic manner explained above, we generate short rules with a small number of conditions in this paper. The number of conditions except for "don't care" conditions is often referred to as the rule length. The maximum rule length is set to three in this paper.

Since a large number of rules are generated even if the maximum rule length is set to three, a pre-screening is performed. We set the minimum confidence level and the minimum support level. From the rules which meet these levels, we select the best 300 rules per class with respect to the product of the confidence and support values. In some cases, the selected rules for the minority class are less than 300. Thus, less than $300M$ rules are selected as the candidate rules at this stage.

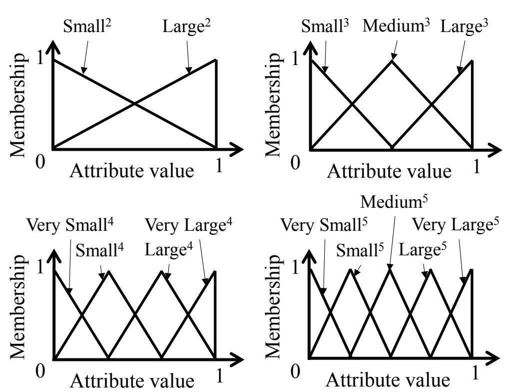

Figure 3: Fuzzy membership functions with multiple granularities for antecedent fuzzy sets.

2.3 Multiobjective Classifier Design

At the final stage, the combination of the candidate rules is optimized by evolutionary computation (EC) in order to obtain classifiers. A classifier is represented by a binary string S of length $300M$. Each bit is corresponding to one of the candidate rules. "1" means that the rule is used for the classifier S, while "0" means that the rule is not used for the classifier S. We consider three objective functions. The first objective function $F_1(S)$ is the accuracy of the classifier S. The second objective function $F_2(S)$ is the number of rules in S. The third objective function $F_3(S)$ is the total rule length of S. This optimization can be formulated as:

$$\text{Maximize } F_1(S), \text{ minimize } F_2(S), \text{ and minimize } F_3(S). \tag{7}$$

For calculating $F_1(S)$, the training patterns are classified by S. We use a winner-take-all method of the following form:

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \mid R_q \in S\}. \tag{8}$$

Each pattern $\mathbf{x}_p$ is classified by the rule which has the maximum product of the compatibility grade and the rule weight. If some rules with different class labels have the same maximum value of (8), this classification is rejected. If no rule matches this pattern, this classification is also rejected. We count the rejection as a misclassification.

In this paper, we use NSGA-II [8] as an optimizer. We use binary tournament selection, uniform crossover and bit-flip mutation as genetic operators.

3 CASE STUDIES

To demonstrate the proposed framework, we apply it to two engineering problems: the welded beam design problem [9] and the conceptual design optimization of hybrid rocket engine [14].

3.1 Parameter Setting

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, $\mathbf{x} = (h, l, t, b)$. The objective functions are the cost of the beam $f_1(\mathbf{x})$ and the vertical deflection at the end of the beam $f_2(\mathbf{x})$ as follows:

$$\text{Minimize } f_1(\mathbf{x}) = 1.10471h^2l + 0.04811tb(14.0 + l), \quad (9)$$

$$\text{Minimize } f_2(\mathbf{x}) = \frac{2.1952}{t^3 b}, \quad (10)$$

$$0.125 \le h, b \le 5.0, \quad (11)$$

$$0.1 \le l, t \le 10.0. \quad (12)$$

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

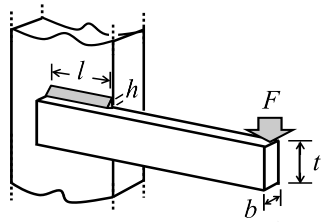

Figure 4: The welded beam design problem.

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.

Design variable	Minimum	Maximum
h	0.171	4.541
l	0.429	9.972
t	2.171	10.000
b	0.238	5.000

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 $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. 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.

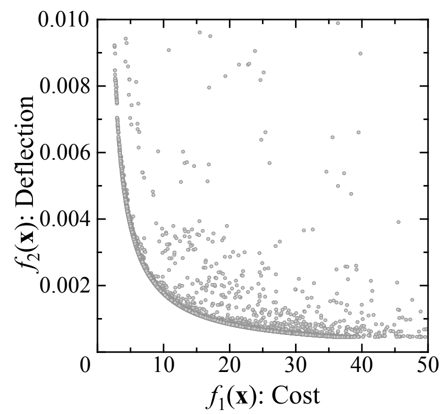

Figure 5: Solutions evaluated by NSGA-II for the welded beam design problem.

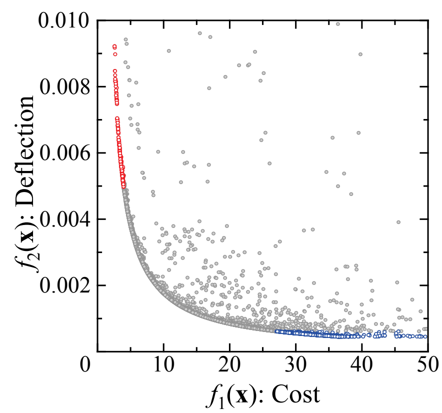

Figure 6: Class specification 1 (two extreme regions) for the welded beam design problem.

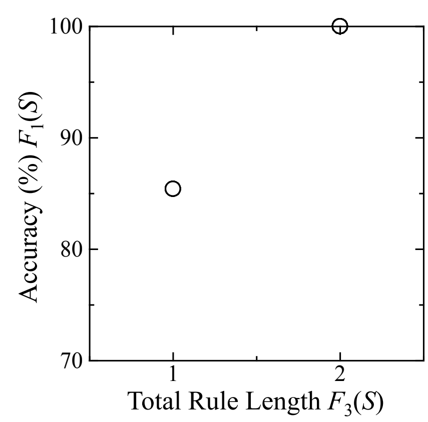

Figure 7: The classifiers obtained by multiobjective classifier design for the class specification 1.

Figure 7 shows the non-dominated classifiers obtained by multiobjective classifier design. The classifiers are projected into the F_1-F_3 space. The most accurate classifier has two rules as:

If b is Small^3 then Class 1 with 1.00,
If b is Large^2 then Class 2 with 0.99.

"Small^3" and "Large^2" 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.

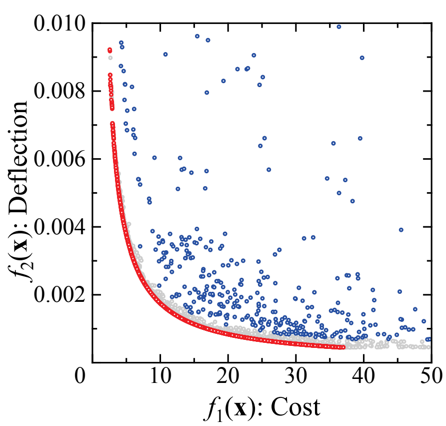

Figure 8: Class specification 2 (Pareto or non-Pareto) for the welded beam design problem.

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^4 then Class 1 with 0.71,
If *t* is Medium^3 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^3 and *b* is Large^2 then Class 1 with 0.74,
If *t* is Small^2 then Class 2 with 1.00,
If *h* is Large^4 then Class 2 with 0.99,
If *t* is Large^5 then Class 2 with 0.99,
If *l* is Medium^3 and *b* is Medium^5 then Class 2 with 0.42,
If *t* is Small^3 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.

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

		Predicted Class		
		Class 1	Class 2	Rejection
True Class	1	1,698	8	0
	2	69	361	0

Table 3: The confusion matrix of the classifier with length 8.

		Predicted Class		
		Class 1	Class 2	Rejection
True Class	1	1,704	2	0
	2	5	425	0

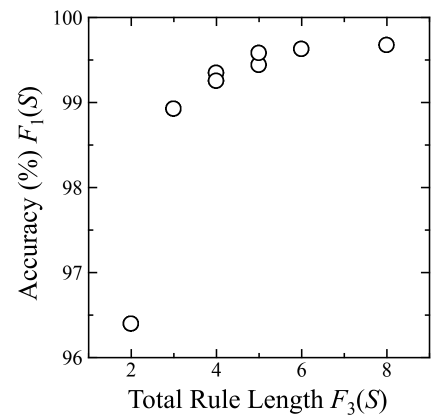

Figure 9: The classifiers obtained by multiobjective classifier design for the class specification 2.

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: http://flab.eng.isas.jaxa.jp/member/oyama/realproblems_j.html.

The conceptual design optimization problem of hybrid rocket engine has mainly six design variables: the mass flow of oxidizer m_{oxi}, the fuel length L_{fuel}, the port radius of fuel r_{port}, the combustion time t_{burn}, the pressure of combustion chamber P_c, and the aperture ratio of nozzle ε. There are two main objective functions: the flight altitude H_{max} $f_1(\mathbf{x})$ and the total gross weight M_{tot} $f_2(\mathbf{x})$. Figure 10 shows the solutions obtained by MOGA in the objective space.

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 $f_1(\mathbf{x})$, the best region on $f_2(\mathbf{x})$, 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).

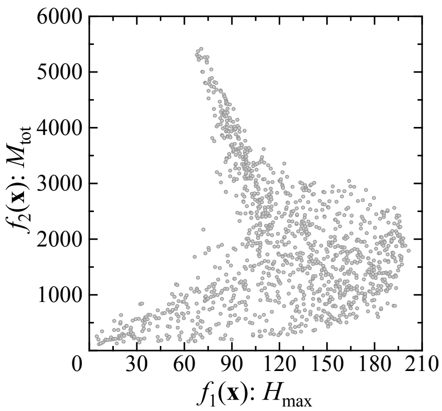

Figure 10: Solutions evaluated by MOGA for the concept design optimization of hybrid rocket engine.

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

Design variable	Minimum	Maximum
m_{oxi}	1.017	30.000
L_{fuel}	1.008	9.998
r_{port}	10.056	199.718
t_{burn}	15.000	34.989
P_c	30.010	39.996
ε	5.000	6.999

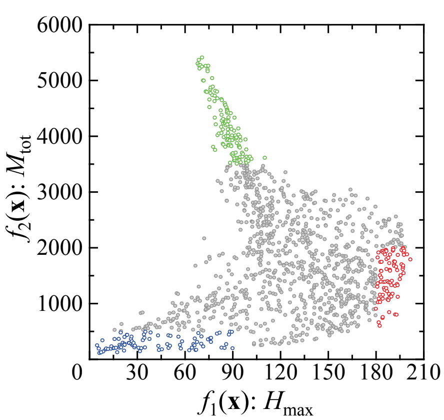

Figure 11: Class specification 3 (three extreme regions) for the conceptual design optimization of hybrid rocket engine.

Figure 12 shows the non-dominated classifiers obtained by multiobjective classifier design. The classifiers with length 1, 2, 3, and 4 have respectively 1, 2, 3 and 3 rules. Since this is the three-class data, at least three rules are necessary in order to correctly classify all patterns. The second most accurate classifier has 98.7% accuracy, while the most accurate classifier has 100% accuracy. The difference was only one condition in the first rule.

The second most accurate classifier was as follows:

If r_{port} is Small^4 then Class 1 with 0.28,
If m_{oxi} is Very Small^4 then Class 2 with 1.00,
If L_{fuel} is Large^3 then Class 3 with 0.76.

The most accurate classifier was as follows:

If r_{port} is Small^4 and P_c is Small^2 then Class 1 with 0.73,
If m_{oxi} is Very Small^4 then Class 2 with 1.00,
If L_{fuel} is Large^3 then Class 3 with 0.76.

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.

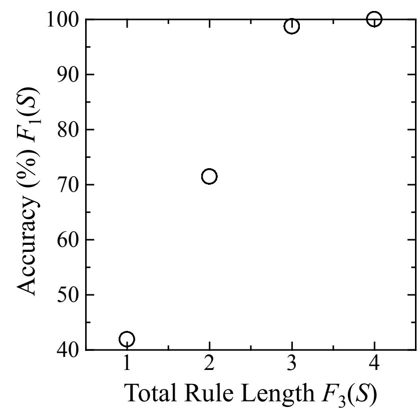

Figure 12: The classifiers obtained by multiobjective classifier design for the class specification 3.

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_{oxi} is Very Small^4 then Class 1 with 0.57,
If L_{fuel} is Large^2 then Class 2 with 0.85.

The accuracy of this classifier was 93.6%. Table 5 shows the confusion matrix. By only two rules, almost all dominated solutions can be correctly classified.

The classifier around the knee point in Fig. 14 is as follows.

If m_{oxi} is Very Small^5 then Class 1 with 0.74,
If L_{fuel} is Small^5 and r_{port} is Small^4 then Class 1 with 0.55,
If L_{fuel} is Large^2 and t_{burn} is Large^2 then Class 2 with 0.91.

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.

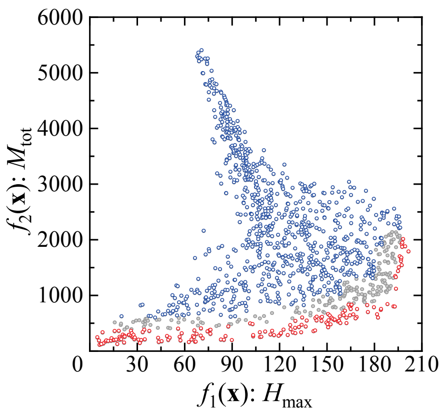

Figure 13: Class specification 4 (Pareto or non-Pareto) for the conceptual design optimization of hybrid rocket engine.

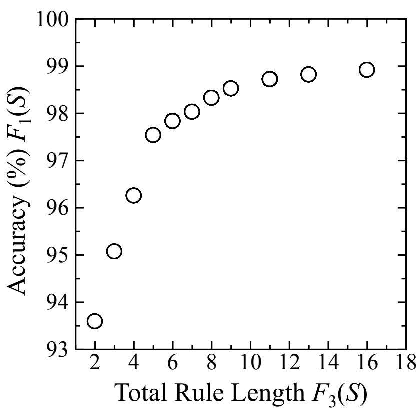

Figure 14: The classifiers obtained by multiobjective classifier design for the class specification 4.

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

		Predicted Class		
		Class 1	Class 2	Rejection
True Class	1	107	63	0
	2	1	842	1

Table 6: The confusion matrix of the classifier with length 5.

		Predicted Class		
		Class 1	Class 2	Rejection
True Class	1	154	16	0
	2	8	835	1

The accuracy of the most accurate classifier was 98.9%. Table 7 shows the confusion matrix. The classifier consists of nine rules and 16 conditions:

If m_{oxi} is Very Small^4 then Class 1 with 0.57,
If m_{oxi} is Very Small^5 then Class 1 with 0.74,
If L_{fuel} is Lage^2 and t_{burn} is Large^2 then Class 1 with 0.91,
If L_{fuel} is Small^5 and r_{port} is Medium^3 then Class 1 with 0.56,
If L_{fuel} is Small^5 and r_{port} is Small^4 then Class 1 with 0.55,
If m_{oxi} is Very Large^5 then Class 2 with 0.98,
If m_{oxi} is Large^3 and ε is Small^2 then Class 2 with 0.97,
If L_{fuel} is Medium^3 and r_{port} is Small^2 and t_{burn} is Large^2 then Class 2 with 0.82,
If m_{oxi} is Large^4 and t_{burn} is Large^3 then Class 2 with 0.97.

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

		Predicted Class		
		Class 1	Class 2	Rejection
True Class	1	159	11	0
	2	0	844	0

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 h is Very Small^4 then Class 1 with 0.61,
If t is Medium^3 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^3 and b is Large^2 then Class 1 with 0.65,
If t is Small^2 then Class 2 with 1.00,
If h is Large^4 then Class 2 with 0.99,
If t is Large^5 then Class 2 with 1.00,
If l is Medium^3 and b is Medium^5 then Class 2 with 0.51,
If h is Small^4 and l is Medium^3 and b is Medium^3 then Class 2 with 0.35,

If l is Medium[3] and b is Large[4] then Class 2 with 0.70,
If l is Medium[3] and b is Large[5] 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.

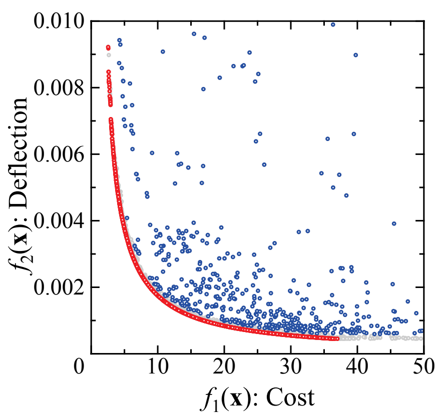

Figure 15: Class specification 5 (Pareto or non-Pareto) for the welded beam design problem.

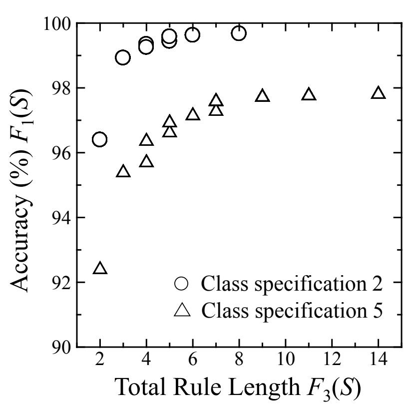

Figure 16: The classifiers obtained by multiobjective classifier design for the class specification 5.

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

[1] R. Alcala, Y. Nojima, F. Herrera, and H. Ishibuchi, “Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions,” *Soft Computing*, vol. 15, no. 12, pp. 2303-2318, November 2011.

[2] S. Bandaru and K. Deb, “Automated innovization for simultaneous discovery of multiple rules in bi-objective problems,” *Evolutionary Multi-Criterion Optimization (EMO2011), LNCS 6576*, Springer, Heidelberg, 2011, pp. 1-15.

[3] S. Bandaru and K. Deb, “Towards automating the discovery of certain innovative design principles through a clustering-based optimization technique,” *Engineering Optimization*, vol. 43, no. 9, pp. 911-941, 2011.

[4] S. Bandaru, A. H. C. Ng, and K. Deb, “Data mining methods for knowledge discovery in multi-objective optimization: Part A – Survey,” *Expert Systems with Applications*, vol. 70, pp. 139-159, 2017.

[5] R. Barandela, J. S. Sanchez, V. Garcia, and E. Rangel, “Strategies for learning in class imbalance problems,” *Pattern Recognition*, vol. 36, pp. 849-851, 2003.

[6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, 2001.

[7] K. Deb, “Unveiling innovative design principles by means of multiple conflicting objectives,” *Engineering Optimization*, vol. 35, no. 5, pp. 445-470, October 2003.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. on Evolutionary Computation*, no. 6, vol. 2, pp. 182-197, April 2002.

[9] K. Deb and A. Srinivasan, “Innovization: Innovating design principles through optimization,” *Proc. of Genetic and Evolutionary Computation Conference*, pp. 1629-1636, 2006.

[10] C. Fonseca and P. Fleming, “Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization,” *Proc. of 5th International Conference on Genetic Algorithms*, pp. 416-423, San Francisco, CA, USA, 1993.

[11] H. Ishibuchi, T. Yamamoto, and T. Nakashima, “Fuzzy data mining: Effect of fuzzy discretization,” *Proc. of 2001 IEEE International Conference on Data Mining*, pp. 241-248, San Jose, USA, November 29 - December 2, 2001.

[12] H. Ishibuchi, Y. Nojima, and I. Kuwajima, “Evolutionary multiobjective design of fuzzy rule-based classifiers,” *In J. Fulcher and L. C. Jain (eds.): Computational Intelligence: A Compendium*, Springer, Berlin, pp. 641-685 (Chapter 15), July 2008.

[13] S. Jeong, K. Chiba, and S. Obayashi, “Data mining for aerodynamic design space,” *Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 11, pp. 452-469, 2005.

[14] Y. Kosugi, A. Oyama, K. Fujii, and M. Kanazaki, “Multidisciplinary and multi-objective design exploration methodology for conceptual design of a hybrid rocket,” *In AIAA Paper 2011-1634*, AIAA (2011).

[15] M. Nakata and K. Chiba, “Design strategy generation for a sounding hybrid rocket via evolutionary rule-based data mining system,” *Proc. of 20th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pp. 305-318, 2016.

[16] S. Obayashi and D. Sasaki, “Visualization and data mining of Pareto solutions using self-organizing map,” *Evolutionary Multi-Criterion Optimization (EMO2003), LNCS 2632*, Springer, Heidelberg, 2003, pp. 796-809.

[17] A. Pyrke, S. Mostaghim, and A. Nazemi, “Heatmap visualization of population based multiobjective algorithms,” *Proc. of 4th International Conference on Evolutionary Multi-criterion Optimization, EMO 2007*, pp. 361-375, 2007.

[18] K. Sugimura, S. Obayashi, and S. Jeong, “A new design method based on cooperative data mining from multi-objective design space,” *Journal of Computational Science and Technology*, vol. 3, no. 1, pp. 287-302, 2009.

[19] S. Watanabe, S. Ohe, and R. Minato, “Development of a multi-granularity design support system using hierarchical clustering for non-dominated solutions,” *Proc. of World Automation Congress*, 6 pages, 2010.