Multiobjectivization from Two Objectives to Four Objectives in Evolutionary Multi-Objective Optimization Algorithms

Hisao Ishibuchi, Yasuhiro Hitotsuyanagi, Yusuke Nakashima, and Yusuke Nojima
Dept. of Computer Science and Intelligent Systems, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka, 599-8531, Japan
{hisaoi@, hitotsu@ci., ysknksm@ci., nojima@}cs.osakafu-u.ac.jp

Abstract— **Multiobjectivization is an interesting idea to solve a difficult single-objective optimization problem through its reformulation as a multiobjective problem. The reformulation is performed by introducing an additional objective function or decomposing the original objective function into multiple ones. Evolutionary multiobjective optimization (EMO) algorithms are often used to solve the reformulated problem. Such an optimization approach, which is called multiobjectivization, has been used to solve difficult single-objective problems in many studies. In this paper, we discuss the use of multiobjectivization to solve two-objective problems. That is, we discuss the idea of solving a two-objective optimization problem by reformulating it as a four-objective one. In general, the increase in the number of objectives usually makes the problem more difficult for EMO algorithms. Thus the handling of two-objective problems as four-objective ones may simply lead to the deterioration in the quality of obtained non-dominated solutions. However, in this paper, we demonstrate through computational experiments that better results are obtained for some two-objective test problems by increasing the number of objectives from two to four.**

Keywords- multiobjectivization; single-objective optimization; evolutionary multiobjective optimization; knapsack problems; fuzzy genetics-based machine learning

I. INTRODUCTION

The term of *"multiobjectivization"* was first proposed by Knowles et al. [1] in 2001. The basic idea is to solve a single-objective optimization problem by reformulating it as a multiobjective problem. Even when the original problem is very difficult with a number of local optima, the reformulated multiobjective problem is not always so difficult. Since 2001, multiobjectivization has been used in many studies to solve difficult single-objective optimization problems [1]-[10]. Multiobjective problems have been generated by introducing additional objectives to single-objective problems. Added objectives are called *"helper-objectives"* [4]. Multiobjective problems have been also generated by decomposing the original objective function into multiple ones [7]. The newly generated multiobjective problems are usually solved by evolutionary multiobjective optimization (EMO) algorithms.

In this paper, we examine the application of the idea of multiobjectivization to solve two-objective problems. That is, a two-objective problem is solved through its reformulation as a four-objective problem. Since EMO algorithms usually do not work well on many-objective problems with four or more objectives [11], our idea is likely to degrade the quality of solutions obtained by EMO algorithms. However, in this paper, we demonstrate that better results are obtained for some two-objective test problems by solving them as four-objective problems rather than directly solving their original two-objective versions.

This paper is organized as follows. First we explain the motivation behind our idea in Section II. Next we show how a four-objective problem is generated from a two-objective one in Section III. Then we shown experimental results on some two-objective test problems in Section IV where we compare obtained solutions between two-objective and four-objective formulations. Finally we conclude this paper in Section V.

II. MOTIVATION

Well-known and frequently-used EMO algorithms such as NSGA-II [12], SPEA [13] and SPEA2 [14] usually work well on multiobjective problems with two or three objectives. In some cases, however, it is difficult for those EMO algorithms to find the entire Pareto front of a large-scale multiobjective combinatorial optimization problem even when the number of objective is two. For example, it has been pointed out in some studies [15]-[18] that only a part of the Pareto front was found by well-known and frequently-used EMO algorithms for large-scale two-objective knapsack problems.

In Fig. 1, we show all individuals at some generations in a single run of NSGA-II [12] on the two-objective 500-item knapsack problem in [13]. We used the following standard settings in Fig. 1 (the same settings were used in other parts of this paper for the same two-objective knapsack problem):

Coding: Binary string of length 500,
Population size: 200,
Initialization: Random initialization,
Constraint handling: Greedy repair in [13],
Crossover probability: 0.8 (uniform crossover),
Mutation probability: 2/500 (bit-flip mutation),
Termination condition: 1,000,000 solution evaluations.

This termination condition means that the execution of NSGA-II was terminated at the 5000th generation since the population size was 200. As shown in Fig. 1, the final population at the 5000th generation is close to the Pareto front. That is, the convergence to the Pareto front is good. However, the diversity along the Pareto front is not enough.

978-1-4244-7376-2/10/$26.00 ©2010 IEEE

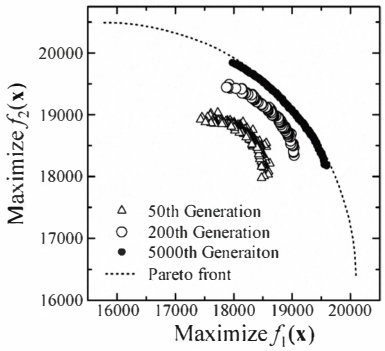


Figure 1. Results of NSGA-II on the two-objective 500-item knapsack problem.

A number of approaches have already been proposed to increase the diversity of solutions in the EMO community. One idea is to use a modified dominance relation. For example, Sato et al. [17] demonstrated that the performance of NSGA-II on two-objective knapsack problems was clearly improved by increasing the diversity of solutions through dominance modification. They also demonstrated that the performance on many-objective knapsack problems was improved by decreasing the diversity of solutions. That is, the diversity should be increased for two-objective problems whereas it should be decreased for many-objective ones.

Our idea of multiobjectivization is to solve two-objective problems after reformulating them as four-objective ones in order to increase the diversity of solutions. At the same time, we try to maintain the high convergence property of NSGA-II on two-objective problems by generating correlated four objectives. In the next section, we explain how to generate four objectives from a two-objective problem in detail.

III. Formulation of Four-Objective Problem

Let us denote a two-objective maximization problem as

$$\text{Maximize } f_1(\mathbf{x}) \text{ and } f_2(\mathbf{x}), \tag{1}$$

$$\text{subject to } \mathbf{x} \in \mathbf{X}, \tag{2}$$

where $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are two objectives to be maximized, $\mathbf{x}$ is the decision vector in the decision space, and $\mathbf{X}$ is the feasible region of $\mathbf{x}$ in the decision space.

We generate the following four-objective problem from the two-objective one in (1)-(2):

$$\text{Maximize } g_1(\mathbf{x}), g_2(\mathbf{x}), g_3(\mathbf{x}) \text{ and } g_4(\mathbf{x}), \tag{3}$$

$$\text{subject to } \mathbf{x} \in \mathbf{X}, \tag{4}$$

where $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, $g_3(\mathbf{x})$ and $g_4(\mathbf{x})$ are defined using a small positive real number α as follows:

$$g_1(\mathbf{x}) = f_1(\mathbf{x}) - \alpha \cdot f_2(\mathbf{x}), \tag{5}$$

$$g_2(\mathbf{x}) = f_1(\mathbf{x}) + \alpha \cdot f_2(\mathbf{x}), \tag{6}$$

$$g_3(\mathbf{x}) = f_2(\mathbf{x}) + \alpha \cdot f_1(\mathbf{x}), \tag{7}$$

$$g_4(\mathbf{x}) = f_2(\mathbf{x}) - \alpha \cdot f_1(\mathbf{x}). \tag{8}$$

Since α is a small positive real number, $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are similar to $f_1(\mathbf{x})$. For the same reason, $g_3(\mathbf{x})$ and $g_4(\mathbf{x})$ are similar to $f_2(\mathbf{x})$. The generated four objectives are weighted sum scalarizing functions of the two objectives $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ with the weight vectors $(1, -\alpha)$, $(1, \alpha)$, $(\alpha, 1)$, $(-\alpha, 1)$, respectively. Each weight vector can be viewed as the search direction of the corresponding weighted sum scalarizing function in the original objective space with $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. In Fig. 2, we show the relation between the original two objectives and the generated four objectives in the original objective space. In this paper, the generated four objectives $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, $g_3(\mathbf{x})$ and $g_4(\mathbf{x})$ are used to solve the original two-objective problem with $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ in Fig. 2.

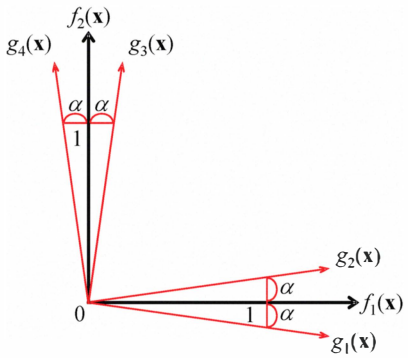


Figure 2. Relation between the original two objectives $f_1(\mathbf{x})$, $f_2(\mathbf{x})$ and the generated four objectives $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, $g_3(\mathbf{x})$, $g_4(\mathbf{x})$ for the case of $\alpha = 0.1$.

IV. Computational Experiments

A. Results on a Two-Objective Knapsack Problem

Using the same parameter settings as in Fig. 1, we applied NSGA-II to the generated four-objective knapsack problem with α = 0.1. In Fig. 3, we show experimental results of a single run in the same manner as in Fig. 1. It is clear from the comparison between Fig. 1 and Fig. 3 that the final solution set at the 5000th generation in Fig. 3 has much large diversity along the Pareto front. Whereas the diversity improvement by the use of the four-objective problem is clear, no clear convergence deterioration is observed in Fig. 3.

In Fig. 4, we show the effect of the value of α on the final results at the 5000th generation. We examined four values of α (α = 0.05, 0.1, 0.2, 0.5). For each value of α, we applied NSGA-II to the corresponding four-objective problem 100 times. Then we calculated the 50% attainment surface. The 50% attainment surface can be viewed as the median over multiple runs (see [19] for details). For comparison, we also applied the NSGA-II to the original two-objective knapsack problem 100 times to calculate the 50% attainment surface. The true Pareto front of the original two-objective knapsack problem is also shown in each plot in Fig. 4. The scale of each plot in Fig. 4 is the same as Fig. 1 and Fig. 3.

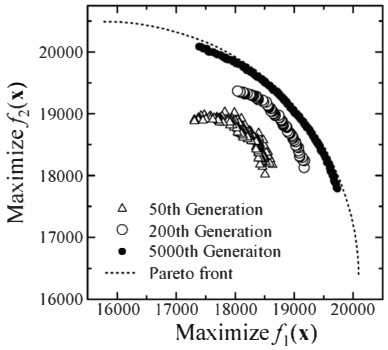


Figure 3. Results of NSGA-II on the four-objective knapsack problem (α = 0.1).

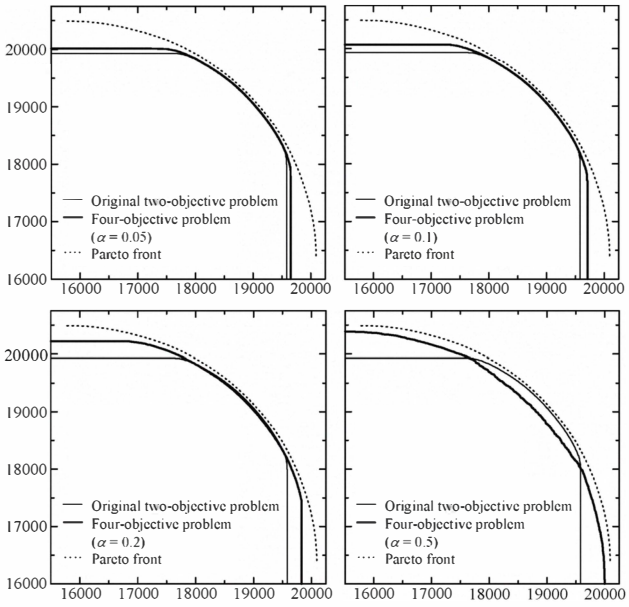


Figure 4. 50% attainment surface for each value of α (α = 0.05, 0.1, 0.2, 0.5) in the four-objective problem together with 50% attainment surface for the original two-objective problem and its true Pareto front. NSGA-II was used.

In Fig. 4, we can see that the increase in the parameter value of α leads to the increase in the diversity of solutions. The deterioration of the convergence toward the Pareto front is not clear when α is not too large (i.e., α = 0.05, 0.1, 0.2). However, in the right-bottom plot of Fig. 4 with α = 0.5, we can observe the tradeoff between the diversity of solutions and their convergence toward the Pareto front. That is, the convergence was degraded while the diversity was improved enough to cover the entire Pareto front when α = 0.5.

In Sato et al. [17], it was shown that the diversity of solutions was increased by modifying the dominance relation for the non-dominated sorting in NSGA-II. The dominance modification for diversity improvement in [17] has the same effect as the use of the following two-objective problem:

$$h_1(\mathbf{x}) = f_1(\mathbf{x}) - \alpha \cdot f_2(\mathbf{x}), \quad (9)$$

$$h_2(\mathbf{x}) = f_2(\mathbf{x}) - \alpha \cdot f_1(\mathbf{x}). \quad (10)$$

These two objectives are the same as $g_1(\mathbf{x})$ and $g_4(\mathbf{x})$ in our four-objective formulation. In Fig. 5, we show experimental results of NSGA-II on the modified two-objective problem. Computational experiments in Fig. 5 were performed in the same manner as in Fig. 4. Similar results were obtained for each value of α in Fig. 4 and Fig. 5. Only when α was specified as α = 0.5, we can observe that somewhat better convergence was obtained from the modified two-objective problem than the four-objective one (compare the 50% attainment surface between Fig. 4 and Fig. 5 for α = 0.5).

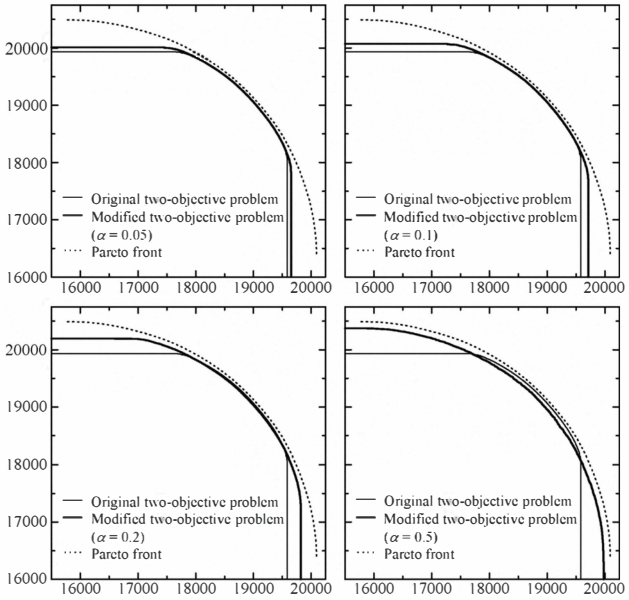


Figure 5. 50% attainment surface for each value of α (α = 0.05, 0.1, 0.2, 0.5) in the modified two-objective problem in (9)-(10). NSGA-II was used.

In the same manner as Fig. 4 for NSGA-II, we also examined the effect of the multiobjectivization from the two-objective problem to the four-objective one for SPEA2 [14]. Experimental results are summarized in Fig. 6. SPEA2 was executed 100 times for each value of α using the same settings as in Fig. 4. Experimental results on the original two-objective problem by SPEA2 are also shown in Fig. 6.

We can see from Fig. 6 that the use of the four-objective problem increased the diversity of solutions along the Pareto front while it slightly degraded the convergence property. Similar effects of the use of the four-objective problem on the 50% attainment surfaces are observed in Fig. 4 with NSGA-II and Fig. 6 with SPEA2. This may be because both NSGA-II and SPEA2 use similar fitness evaluation mechanisms based on the Pareto dominance relation.

We also examined the effect of the multiobjectivization from the two-objective problem to the four-objective problem for MOEA/D [20]. MOEA/D is a recently-developed high-performance scalarizing function-based EMO algorithm. This algorithm does not use the Pareto dominance relation for fitness evaluation. Thus the behavior of MOEA/D is usually totally different from NSGA-II and SPEA2.

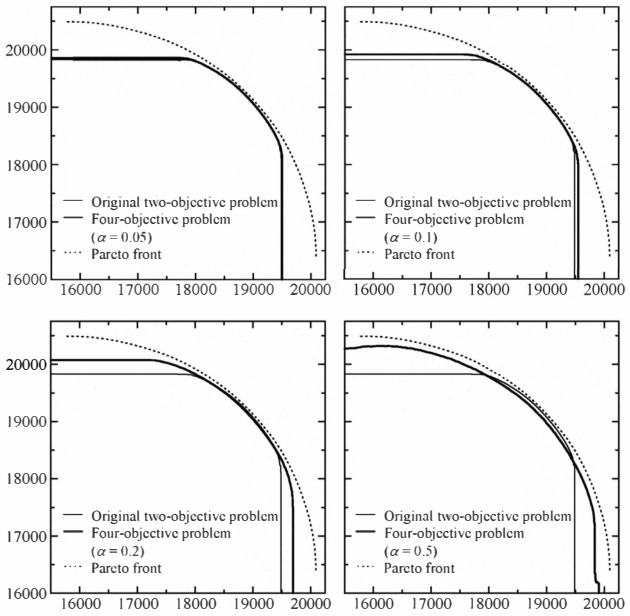


Figure 6. 50% attainment surface for each value of α (α = 0.05, 0.1, 0.2, 0.5) in the four-objective problem together with 50% attainment surface for the original two-objective problem and its true Pareto front. SPEA2 was used.

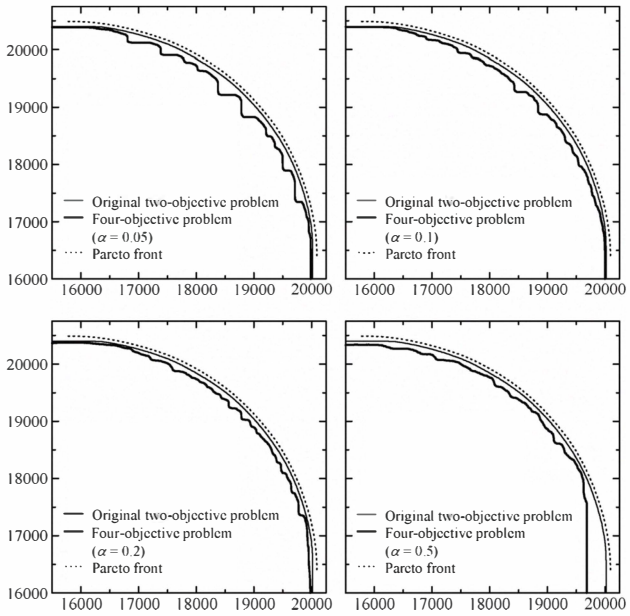
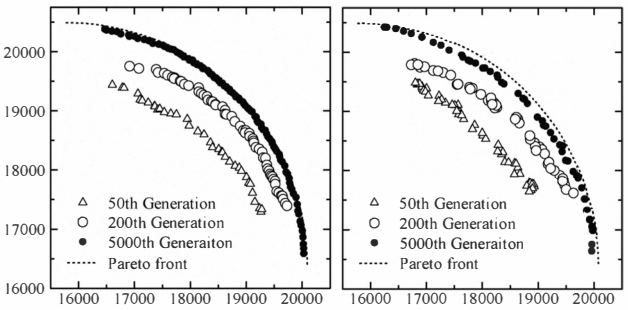


Figure 7. 50% attainment surface for each value of α (α = 0.05, 0.1, 0.2, 0.5) in the four-objective problem together with 50% attainment surface for the original two-objective problem and its true Pareto front. MOEA/D was used.

In Fig. 7, experimental results by MOEA/D are shown in the same manner as in Fig. 4 and Fig. 6. We used the weighted Tchebycheff function in MOEA/D. The population size in MOEA/D was specified as 200 for the two-objective problem and 220 for the four-objective problem. Since the population size in MOEA/D is determined by the number of uniformly distributed weight vectors (for details, see [20]), we used the different specification of the population size only for MOEA/D on the four-objective problem. Except for the population size, we used the same settings in Figs. 4-6.

In Fig. 7, good results were obtained from MOEA/D on the original two-objective problem. MOEA/D on the original two-objective problems outperformed NSGA-II and SPEA2. The use of the four-objective problem leads to strange shapes of the attainment surfaces. This is because the granularity of weight vectors becomes coarse by the increase in the number of objectives. For examining the behavior of MOEA/D, we show experimental results of its single run on the original two-objective and formulated four-objective problems in Fig. 8. In the case of the four-objective problem in Fig. 8 (b), a large number of solutions were not obtained along the Pareto front since the granularity of the weight vectors is coarse.

For quantitative examination, we calculate the average value of the hypervolume over 100 runs of NSGA-II for various specifications of α in the four-objective and modified two-objective problems (see [19] for various performance measures of EMO algorithms including hypervolume). The hypervolume was calculated in the objective space of the original two-objective problem using the origin (0, 0) as the reference point. Experimental results are summarized in Table I. Similar performance improvement of NSGA-II was achieved by the four-objective problem formulation and the two-objective problem modification. When the value of α was close or equal to 1.0, good results were not obtained.



(a) Original two-objective problem. (b) Four-objective problem (α=0.1).

Figure 8. Experimental results on a single run of MOEA/D on the two-objective problem and the four-objective problem.

TABLE I. AVERAGE HYPERVOLUME VALUES AND STANDARD DEVIATIONS BY NSGA-II

Problem	Generated Four-Objective Problem						Original
α	0.1	0.2	0.4	0.6	0.8	1.0	-
Ave. ($\times 10^8$)	3.937	3.985	4.035	4.017	3.885	3.355	3.890
SD ($\times 10^6$)	1.498	0.936	0.432	0.521	1.124	3.682	1.604

Problem	Modified Two-Objective Problem						Original
α	0.1	0.2	0.4	0.6	0.8	1.0	-
Ave. ($\times 10^8$)	3.937	3.979	4.031	4.028	3.943	2.770	3.890
SD ($\times 10^6$)	1.376	1.016	0.584	0.472	0.970	4.960	1.604

B. Results on a Two-Objective Fuzzy System Design Problem

We also examined the effectiveness of the proposed idea for a two-objective problem formulation of multiobjective fuzzy genetics-based machine learning [21]. The task is to search for non-dominated fuzzy classifiers with respect to the following two-objectives:

$f_1(S)$: The number of fuzzy rules in a fuzzy classifier S,

$f_2(S)$: The error rate in percentage on training patterns by S.

The first objective measures the complexity of the fuzzy rule-based classifier S while the second objective is related to the classification accuracy of S. These two objectives are to be minimized to find simple and accurate fuzzy classifiers.

A fuzzy classifier for an n-dimensional classification problem is a set of fuzzy rules of the following type [21]:

$$R_j: \text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then Class } C_j \text{ with } CF_j, \quad (11)$$

where R_j the label of the j-th fuzzy rule, $\mathbf{x} = (x_1, \ldots, x_n)$ is an n-dimensional pattern vector, A_{ji} is an antecedent fuzzy set such as *"small"* and *"large"*, C_j is a consequent class, and CF_j is a rule weight (i.e., a kind of certainty factor).

In our computational experiments, we used 27 triangular fuzzy sets in Fig. 9 and *"don't care"* as antecedent fuzzy sets in fuzzy rules of the form in (11). Thus the total number of combinations of antecedent fuzzy sets for an n-dimensional classification problem is $(27+1)^n$. This is also the number of possible fuzzy rules because the consequent class C_j and the rule weight CF_j of each fuzzy rule are determined from training patterns when its antecedent part is specified [21]. The fuzzy classifier design is to find only a small number of fuzzy rules from such a large number of possible rules.

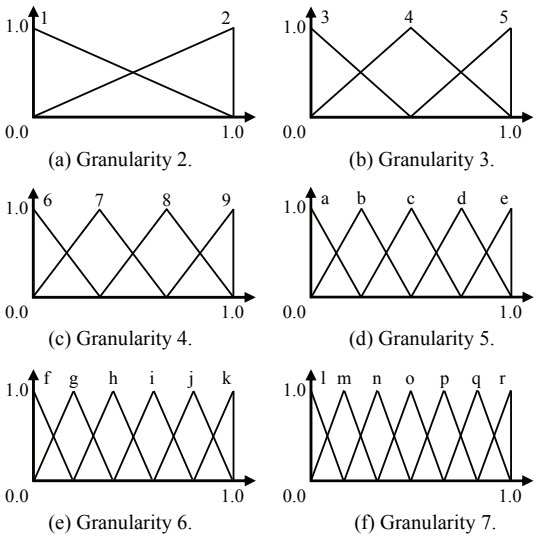


Figure 9. Antecedent fuzzy sets used in our computational experiments.

Due to the page limitation as a conference paper, we do not explain our multiobjective fuzzy genetics-based machine learning algorithm [21] in detail. We only briefly explain its basic outline here. NSGA-II [12] is used as the framework for multiobjective optimization in our algorithm. The search for good rule sets is performed in the framework of Pittsburgh approach where an individual is a set of fuzzy rules. Each fuzzy rule is represented by an integer substring of length n, which shows its n antecedent fuzzy sets. Thus an individual including m fuzzy rules is represented by an integer string of length $m \cdot n$. Since the number of fuzzy rules should be minimized, the string length is not fixed. Its upper and lower bounds, however, are pre-specified (i.e., the maximum and minimum numbers of fuzzy rules in each classifier are fixed). Our crossover operator randomly chooses some fuzzy rules from two parents to generate a new offspring. The crossover operator is used with a pre-specified crossover probability. After crossover, each antecedent fuzzy set in the generated offspring is randomly changed to a different antecedent fuzzy set with a pre-specified mutation probability. A heuristic rule generation mechanism and a Michigan-style rule generation mechanism are also used in our algorithm [21].

Our algorithm was applied to the glass data set in the KEEL (http://keel.es). This data set is a six-class pattern classification problem with 214 patterns and nine attributes (i.e., $n = 9$ in (11)). We used the following settings:

Population size: 200,
Termination condition: 5000 generations,
Number of fuzzy rules in initial individuals: 20,
Upper limit on the number of fuzzy rules: 40,
Lower limit on the number of fuzzy rules: 1.

The ten-fold cross-validation procedure was iterated 10 times (i.e., 10CV × 10) for calculating the average error rates on training and test patterns. This means that our algorithm was executed 100 times using different 90% patterns as training data and the remaining 10% patterns as test data. In each run of our algorithm, multiple fuzzy classifiers with different numbers of fuzzy rules were obtained. The obtained fuzzy classifiers were divided into different groups according to the number of fuzzy rules. Then the average error rate was calculated in each group over those classifiers with the same number of fuzzy rules. Whereas small classifiers with a few fuzzy rules were always obtained, large fuzzy classifiers with many rules were not obtained in many runs. Thus the average error rates were not always calculated over 100 classifiers.

Experimental results on the original two-objective problem are shown in Fig. 10. The radius of each circle is proportional to the number of fuzzy classifiers over which the average error rate was calculated. When the error rate was calculated over 100 fuzzy classifiers (i.e., over the 100 runs), a small circle is added at the center of a large circle. Closed black circles show average error rates on training patterns, and open white circles show those on test patterns.

For comparison, experimental results on the four-objective problem (α = 0.1) are summarized in Fig. 11. Much better results with lower error rates were obtained in Fig. 11 from the four-objective problem.

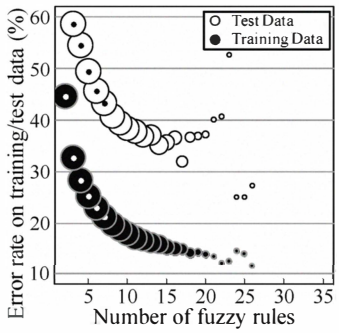


Figure 10. Experimental results of 10×10CV on the glass identification data set. Our multiobjective genetics-based machine learning algorithm was applied to the original two-objective problem.

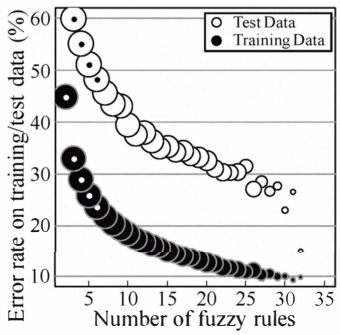


Figure 11. Experimental results of 10×10CV on the glass identification data set. Our multiobjective genetics-based machine learning algorithm was applied to the generated four-objective problem with $\alpha = 0.1$.

V. Concluding Remarks

In this paper, we examined the multiobjectivization from two-objective problems to four-objective problems. Since EMO algorithms often do not work well on many-objective problems with four or more objectives, it was expected that the multiobjectivization to four-objective problems would simply degrade the quality of the obtained solutions by EMO algorithms. However, as demonstrated in this paper, better results with respect to the diversity of solutions were obtained from four-objective formulations for some test problems. Moreover, the convergence of solutions to the Pareto front was not severely degraded by the use of four-objective problems for those test problems. This is because generated four objectives are not independent but correlated with each other. Of course, the use of four-objective problems does not always lead to the improvement of obtained solutions as shown through computational experiments on MOEA/D.

This work was supported in part by the Japan Society for the Promotion of Science under Grant-in-Aid for Scientific Research (B) (20300084).

References

[1] J. D. Knowles, R. A. Watson, D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," *Lecture Notes in Computer Science, Vol. 1993: Evolutionary Multi-Criterion Optimization - EMO 2001*, pp. 269-283, Springer, Berlin, March 2001.

[2] J. Scharnow, K. Tinnefeld, I. Wegener, "Fitness landscapes based on sorting and shortest paths problems," *Lecture Notes in Computer Science. Vol. 2439: Parallel Problem Solving from Nature - PPSN VII*, pp. 54-63, Springer, Berlin, September 2002.

[3] M. T. Jensen, "Guiding single-objective optimization using multi-objective methods," *Lecture Notes in Computer Science, vol. 2611: EvoWorkshops*, pp. 268-279, Springer, January 2003.

[4] M. T. Jensen, "Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation," *Journal of Mathematical Modelling and Algorithms* 3 (4), pp. 323-347, December 2004.

[5] S. Watanabe, K. Sakakibara, "Multi-objective approaches in a single-objective optimization environment," *Proc. of 2005 IEEE Congress on Evolutionary Computation*, pp. 1714-1721, Edinburgh, UK, September 2-4, 2005.

[6] H. Ishibuchi, Y. Nojima, "Optimization of scalarizing functions through evolutionary multiobjective optimization," *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, pp. 51-65, Springer, Berlin, March 2007.

[7] J. Handl, S. C. Lovell, J. Knowles, "Multiobjectivization by decomposition of scalar cost functions," *Lecture Notes in Computer Science. Vol. 5199: Parallel Problem Solving from Nature - PPSN X*, pp. 31-40, Springer, Berlin, September 2008.

[8] J. Handl, S. C. Lovell, J. Knowles, "Investigations into the effect of multiobjectivization in protein structure prediction," *Lecture Notes in Computer Science. Vol. 5199: Parallel Problem Solving from Nature - PPSN X*, pp. 702-711, Springer, Berlin, September 2008.

[9] M. Jähne, X. Li, J. Branke, "Evolutionary algorithms and multi-objectivization for the travelling salesman problem," *Proc. of 2009 Genetic and Evolutionary Computation Conference*, pp. 595-602, Montreal, Canada, July 8-12, 2009.

[10] D. F. Lochtefeld, F. W. Ciarallo, "Deterministic helper-objective sequences applied to job-shop scheduling," *Proc. of 2010 Genetic and Evolutionary Computation Conference*, pp. 431-438, Portland, USA, July 7-11, 2010.

[11] H. Ishibuchi, N. Tsukamoto, Y. Nojima, "Evolutionary many-objective optimization: A short review," *Proc. of 2008 IEEE Congress on Evolutionary Computation*, pp. 2424-2431, Hong Kong, June 1-6, 2008.

[12] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation* 6 (2), pp. 182-197, April 2002.

[13] E. Zitzler, L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation* 3 (4), pp. 257-271, November 1999.

[14] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-Report* 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, 2001.

[15] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment," *IEEE Trans. on Evolutionary Computation* 6 (4), pp. 402-412, August 2002.

[16] A. Jaszkiewicz, "On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study," *European Journal of Operational Research* 158 (2), pp. 418-433, October 2004.

[17] H. Sato, H. E. Aguirre, K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, pp. 5-20, Springer, Berlin, March 2007.

[18] H. Ishibuchi, K. Narukawa, N. Tsukamoto, Y. Nojima, "An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization," *European Journal of Operational Research* 188 (1), pp. 57-75, July 2008.

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

[20] Q. Zhang, H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation* 11 (6), pp. 712-731, December 2007.

[21] H. Ishibuchi, Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *International Journal of Approximate Reasoning* 44 (1), pp. 4-31, January 2007.