Effects of Discrete Objective Functions with Different Granularities on the Search Behavior of EMO Algorithms

Hisao Ishibuchi, Masakazu Yamane and Yusuke Nojima
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
Phone: +81-72-254-9350
{hisaoi@, masakazu.yamane@ci., nojima@}cs.osakafu-u.ac.jp

ABSTRACT
Objective functions in combinatorial optimization are discrete. The number of possible values of each discrete objective function is totally different from problem to problem. Optimization of a discrete objective function is often very difficult. In the case of multiobjective optimization, a different objective function has a different number of possible values. This means that each axis of the objective space has a different granularity. Some axes may have fine granularities while others are coarse. In this paper, we examine the effect of discrete objective functions with different granularities on the search behavior of EMO (evolutionary multiobjective optimization) algorithms through computational experiments. Experimental results show that a discrete objective function with a coarse granularity slows down the search of EMO algorithms along that objective. An interesting observation is that such a slow-down along one objective often leads to the speed-up of the search along other objectives. We also show that the search of some EMO algorithms on test problems is often improved by adding a small random noise to each discrete objective.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic Methods.*

General Terms
Algorithms.

Keywords
Evolutionary multiobjective optimization (EMO), combinatorial optimization, discrete objective functions, genetic algorithms.

1. INTRODUCTION
In the implementation of evolutionary multiobjective optimization (EMO) algorithms [4]-[6], objective values are often normalized so that each objective function has the same range of objective values. The normalization is important in many EMO algorithms when the ranges of objective functions are totally different from each other. This is because fitness evaluation in EMO algorithms often includes distance calculations in the objective space. For example, Pareto dominance-based EMO algorithms such as NSGA-II [7] and SPEA2 [21] have a distance-based secondary criterion in addition to a Pareto dominance-based main criterion. Scalarizing function-based EMO algorithms such as MOEA/D [18] more directly depends on distance calculations. Whereas the necessity of the normalization seems to have been well-known in the EMO community (e.g., NSGA-II [7] has a normalization mechanism, and the normalization was mentioned in [18]), much attention has not been given to the handling of the difference in granularities of discrete objective functions. The goal of this paper is to clearly demonstrate how the difference in the granularities affects the search behavior of EMO algorithms.

Objective functions in combinatorial optimization are discrete (see [1], [3], [16] for various combinatorial optimization problems, their difficulties, and their optimization algorithms). The number of possible values is totally different from problem to problem. Some objectives such as the tour length in travelling salesman problems and the profit in knapsack problems have a large number of possible values. However, other objectives have only a small number of possible values. For example, the number of selected features in feature selection with 30 features has only 31 possible values including 0 to 30 whereas there exist more than one billion (i.e., 2^{30}) combinations of feature selection.

Optimization of a discrete objective function with a small number of possible values is often very difficult for search algorithms. This is because many different solutions have the same objective value (i.e., because the landscape of the objective function is like a stairway with a small number of large flat stairs). For example, local search will not work well when all neighbors of the current solution have the same objective value. When all solutions in the current population have the same fitness, it is difficult for genetic algorithms to efficiently generate promising solutions from the current population. Thus the ability of search algorithms is likely to be deteriorated when they are applied to a discrete objective function with a small number of possible values.

When combinatorial optimization involves multiple objectives, each discrete objective function has a different number of possible values. Let us assume that the above-mentioned feature selection problem with 30 features has the following two objectives: to minimize the error rate and the number of selected features. We also assume that 10,000 training patterns are given to calculate the error rate. In this case, the error rate has 10,001 possible values (from 0% to 100%) whereas the number of selected features has only 31 possible values (from 0 to 30). This means that each axis

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.

of the objective space has a totally different granularity: One axis has a fine granularity with 10,001 possible values while the other has a coarse granularity with only 31 possible values.

The existence of an axis with a coarse granularity (i.e., a discrete objective function with a small number of possible values) will deteriorate the search ability of EMO algorithms along that axis for the above-mentioned reason (i.e., because many different solutions have the same objective value). Such an axis with a coarse granularity may also have a negative effect on the overall search ability of EMO algorithms since it decreases the number of different non-dominated objective vectors in the objective space. In this paper, we examine these effects of discrete objective functions with different granularities on the search behavior of EMO algorithms. Computational experiments are performed using discrete objective functions with a wide range of granularities.

This paper is organized as follows. In Section 2, we explain our test problems with a wide range of granularities. In Section 3, we report experimental results on each test problem. We use the following four EMO algorithms in computational experiments: NSGA-II [7], SPEA2 [21], MOEA/D [18] and SMS-EMOA [2]. NSGA-II and SPEA2 are well-known frequently-used Pareto dominance-based EMO algorithms. MOEA/D can be viewed as a representative of scalarizing function-based EMO algorithms. SMS-EMOA can be a representative of indicator-based EMO algorithms [19], [20]. Its high search ability was demonstrated in [17]. Finally, this paper is concluded in Section 4.

2. TEST PROBLEMS

We generate six test problems with different granularities from a two-objective 500-item 0/1 knapsack problem in Zitzler and Thiele [22]. Multi-objective 0/1 knapsack problems in [22] have been frequently used in the literature to evaluate the search ability of EMO algorithms for combinatorial multiobjective optimization (e.g., Jaszkiewicz [12], [13]). In Fig. 1, we show the Pareto front of the two-objective 500-item 0/1 knapsack problem together with an initial population of randomly generated 200 solutions. Since the initial population is not close to the Pareto front, a high convergence ability is needed to efficiently search for Pareto optimal solutions. At the same time, a diversity improvement ability is also needed since the width of the initial solution is smaller than that of the Pareto front. That is, both the convergence and the diversity are important for EMO algorithms to efficiently search for a good solution set that well approximates the entire Pareto front of the two-objective 500-item 0/1 knapsack problem.

As shown in Fig. 1, let $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ be the objective functions of the two-objective 500-item 0/1 knapsack problem. In [22], $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ were explained as being the total profit for each of the two knapsacks. The minimum possible difference in objective values of each objective function is 1 because the profit of each item is given as an integer in [22]. Thus each objective function can be viewed as having the discretization interval of size 1 (i.e., width 1). We use the size of the discretization interval as an index of the granularity of each objective function. Using this index, we refer to the two-objective 500-item 0/1 knapsack problem in [22] as the G1-G1 2-500 knapsack problem where G1 means the granularity with the discretization interval of size 1.

We generate six test problems by rounding up the first or second digit of objective values of $f_1(\mathbf{x})$ and/or $f_2(\mathbf{x})$. For example, let us assume that the first digit of objective values of $f_1(\mathbf{x})$ is rounded up (e.g., all objective values in the interval [19321, 19330] are changed to 19330). In this case, the minimum possible difference in objective values of $f_1(\mathbf{x})$ is 10 while that of $f_2(\mathbf{x})$ is 1 (because we do not round up objective values of $f_2(\mathbf{x})$). We refer to the generated test problem as the G10-G1 knapsack problem using the granularity G10 of $f_1(\mathbf{x})$ and G1 of $f_2(\mathbf{x})$. In another test problem, the first and second digits of objective values of $f_1(\mathbf{x})$ are rounded up (e.g., all objective values in the interval [19301, 19400] are changed to 19400). Since the minimum possible difference in objective values of $f_1(\mathbf{x})$ is 100, the generated test problem is referred to as the G100-G1 knapsack problem. In this manner, we generate the following six test problems from the original G1-G1 knapsack problem: the G10-G1, G100-G1, G1-G10, G1-G100, G10-G10 and G100-G100 knapsack problems.

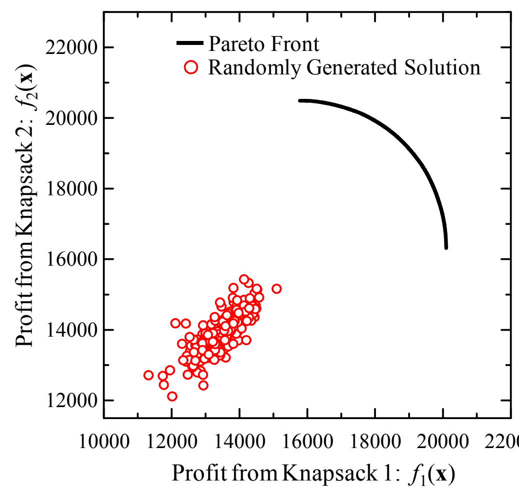

Figure 1. Pareto front and randomly generated 200 solutions.

For visually illustrating the difference between the original G1-G1 problem and the generated ones, we show the Pareto optimal solutions of the G100-G1 and G1-G100 problems together with those of the G1-G1 problem in Fig. 2 and Fig. 3, respectively. From these figures, we can see that the number of different Pareto optimal solutions in the objective space (i.e., different Pareto optimal objective vectors) is decreased by the use of a coarse granularity for one objective. In Fig. 2, our G100-G1 problem has no Pareto optimal solutions around the bottom-right edge of the Pareto front of the original G1-G1 problem. The distribution of different Pareto optimal objective vectors of the G100-G1 problem is sparse in the bottom-right area of Fig. 2 (i.e., this problem does not have many different Pareto optimal objective vectors around the best solution of $f_1(\mathbf{x})$ with the very coarse granularity). From this observation, one may expect difficulties in searching for good solutions in the bottom-right area of the objective space (i.e., good solutions with large values of $f_1(\mathbf{x})$).

On the other hand, in Fig. 3, the distribution of different Pareto optimal objective vectors of our G1-G100 test problem is sparse around the top-left edge of the Pareto front of the original G1-G1 problem. From this observation, one may expect difficulties in searching for good solutions in the top-left area of the objective space (i.e., good solutions with large values of $f_2(\mathbf{x})$).

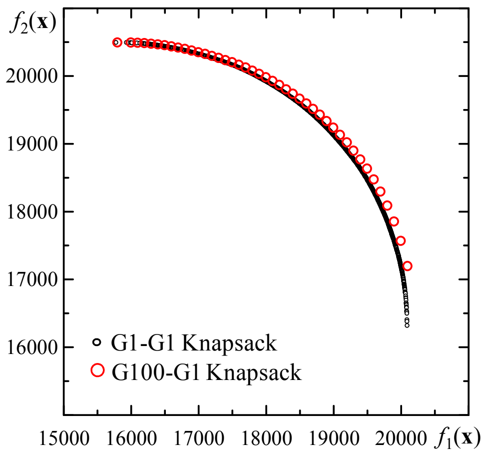

Figure 2. Pareto optimal solutions of the G1-G1 and G100-G1.

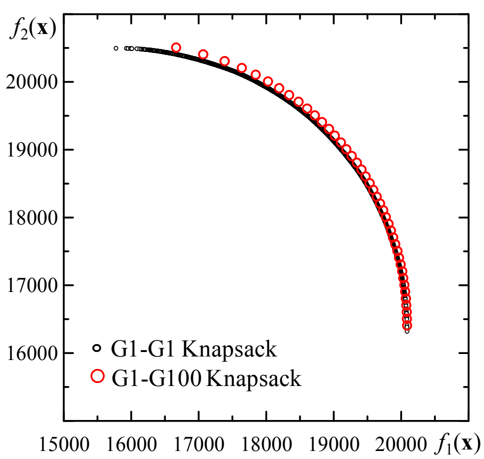

Figure 3. Pareto optimal solutions of the G1-G1 and G1-G100.

The biased distribution of the Pareto optimal objective vectors of each test problem in Fig. 2 and Fig. 3 is due to the convex shape of the Pareto front of the original G1-G1 knapsack problem. For examining test problems with a different shape of the Pareto front, we use a 500-bit one-max zero-max problem. This problem has two objectives to be maximized: the number of "1" and the number of "0" in a binary string of length 500. That is, $f_1(\mathbf{x})$ is the number of "1" while $f_2(\mathbf{x})$ is the number of "0". All binary strings of length 500 are Pareto optimal. In the objective space, this problem has 501 different Pareto optimal objective vectors uniformly distributed on the line between (0, 500) and (500, 0). The shape of the Pareto front of the one-max zero-max problem is a straight line. We refer to this problem as the linear G1-G1 one-max zero-max problem where "linear" means the linear Pareto front. In the same manner as our six knapsack problems (i.e., by rounding up the first or second digit of objective values), we generate the following six problems from the linear G1-G1 one-max zero-max problem: the linear G10-G1, G100-G1, G1-G10, G1-G100, G10-G10 and G100-G100 one-max zero-max problems.

For examining convex and concave Pareto fronts, we generate the following two test problems from the original 500-bit one-max zero-max problem with $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$:

Convex Problem: $g_i(\mathbf{x}) = 500\sqrt{f_i(\mathbf{x})/500},\ i = 1, 2\ ,$ (1)

Concave Problem: $h_i(\mathbf{x}) = 500(f_i(\mathbf{x})/500)^2,\ i = 1, 2\ .$ (2)

We refer to the generated two test problems as the convex and concave G1-G1 one-max zero-max problems, respectively. From each test problem, we generate six problems in the same manner as our six knapsack problems and six linear one-max zero-max problems (e.g., the convex G10-G1, G100-G1, G1-G10, G1-G100, G10-G10 and G100-G100 one-max zero-max problems are generated by rounding up $g_1(\mathbf{x})$ and/or $g_2(\mathbf{x})$ of the convex G1-G1 one-max zero-max problem in (1)).

In Fig. 4, we show the Pareto optimal objective vectors of the convex G1-G1 and G10-G1 one-max zero-max problems. We can observe similar distributions of the Pareto optimal objective vectors in Fig. 2 and Fig. 4. For comparison, we show the Pareto optimal objective vectors of the concave G1-G1 and G10-G1 one-max zero-max problems in Fig. 5. Since the shape of the Pareto front is different between Fig. 4 and Fig. 5, the distribution of their Pareto optimal objective vectors is different.

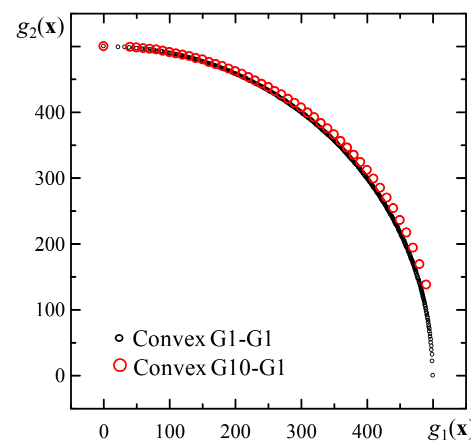

Figure 4. Pareto optimal solutions of the convex G1-G1 and G10-G1 one-max zero-max problems.

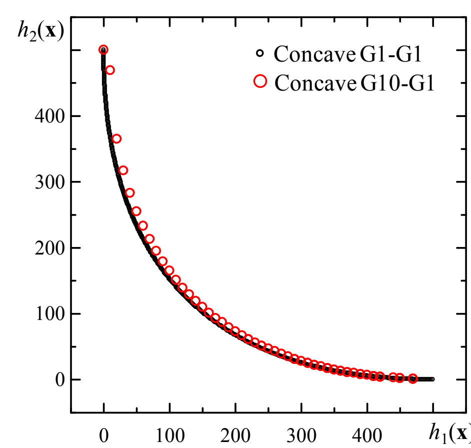

Figure 5. Pareto optimal solutions of the concave G1-G1 and G10-G1 one-max zero-max problems.

3. EXPERIMENTAL RESULTS

3.1 Setting of Computational Experiments

We apply NSGA-II, SPEA2, MOEA/D and SMS-EMOA to our test problems using the following parameter specifications:

Coding: Binary string of length 500 (i.e., 500-bit string),
Population size: 200,
Termination condition: 2000 generations,
Parent selection: Binary tournament selection with replacement,
Crossover: Uniform crossover (Probability: 0.8),
Mutation: Bit-flip mutation (Probability: 1/500),
Number of runs for each test problem: 100 runs.

The 50% attainment surface is calculated over 100 runs of each algorithm for each test problem. In SMS-EMOA, (0, 0) is used as a reference point for hypervolume calculation. In MOEA/D, the weighted Tchebycheff function with the following reference point $\mathbf{z}^*$ is used as a scalarizing function:

$$\text{Knapsack: } z_i^* = 1.1 \cdot \max\{f_i(\mathbf{x}) \mid \mathbf{x} \in \Omega(t)\},\ i = 1, 2, \tag{3}$$

$$\text{One-Max Zero-Max: } z_i^* = \max\{f_i(\mathbf{x}) \mid \mathbf{x} \in \Omega(t)\},\ i = 1, 2, \tag{4}$$

where $\Omega(t)$ shows the population at the t-th generation. The same scalarizing function was used in Zhang and Li [18].

3.2 Results on Knapsack Problems

In Fig. 6, we show experimental results of a single run of NSGA-II on the G1-G1 and G100-G1 knapsack problems. All solutions at the 20th, 50th and 2000th generations are shown in Fig. 6. We can see from Fig. 6 that the use of the very coarse granularity G100 for $f_1(\mathbf{x})$ in the G100-G1 problem biases the search towards the top-left area of the objective space as expected from Fig. 2.

In Fig. 7, we show the 50% attainment surface over 100 runs of each EMO algorithm for the G1-G1, G10-G1 and G100-G1 problems using the "rounded-up" objective functions of each test problem. The effect of the coarse granularity G10 (i.e., the G10-G1 problem) on each EMO algorithm is very small. However, the use of the much coarser granularity G100 (i.e., the G100-G1 problem) clearly biases the search by all the four EMO algorithms towards the top-left area of the objective space in Fig. 7.

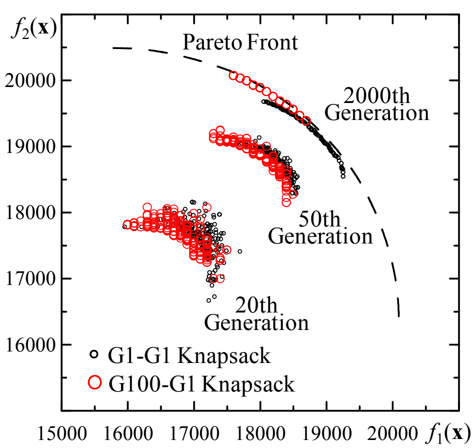

Figure 6. Experimental results of a single run of NSGA-II on the G1-G1 and G100-G1 knapsack problems.

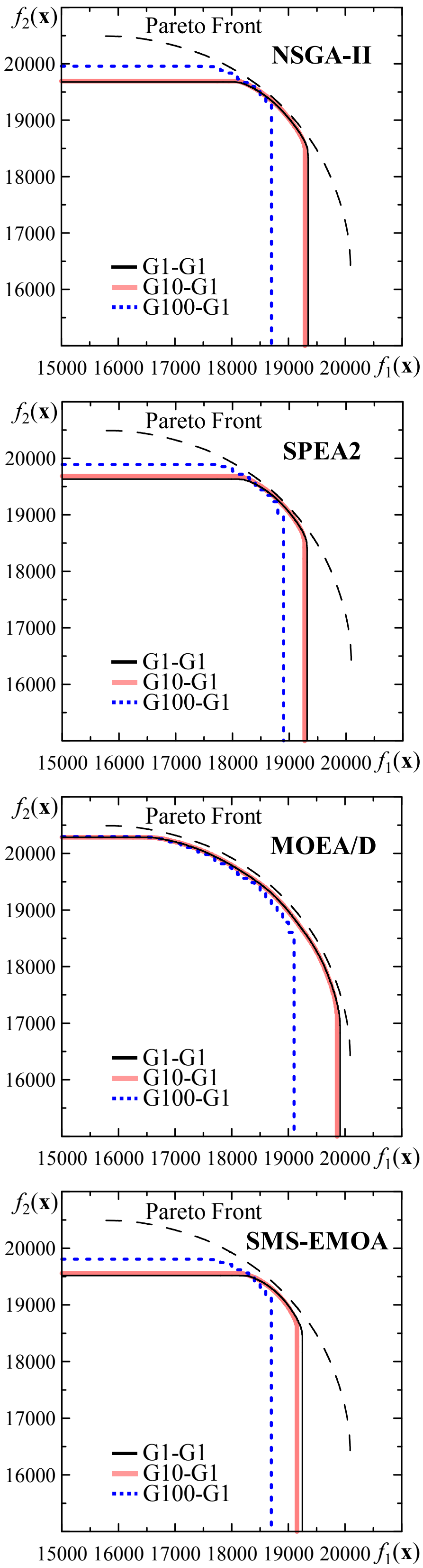

Figure 7. 50% attainment surface of each algorithm for the G1-G1, G10-G1 and G100-G1 knapsack problems.

For comparison, we show experimental results of a single run of NSGA-II on the G1-G100 knapsack problem with the very coarse granularity G100 for the second objective $f_2(\mathbf{x})$ in Fig. 8. We can see from Fig. 8 that the use of the very coarse granularity G100 in $f_2(\mathbf{x})$ biases the search by NSGA-II towards the bottom-right area of the objective space as expected from Fig. 3.

In Fig. 9, we show experimental results of NSGA-II on the G100-G100 problem with the very coarse granularity G100 for both $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. As shown in Fig. 9, the use of the very coarse granularity G100 in both of the two objectives clearly degrades the performance of NSGA-II. This is because many different solutions have the same objective value, which makes it very difficult to efficiently search for good solutions (as we have already mentioned in this paper). Another possible reason is the decrease in the number of different Pareto optimal objective vectors. To examine this issue, we perform computational experiments on the G100-G100 problem and its noisy version where a small noise (a real random number in the interval [0, 1]) is added to each of the calculated values of $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. Experimental results are summarized in Fig. 10 where the 50% attainment surface over 100 runs of each algorithm is shown.

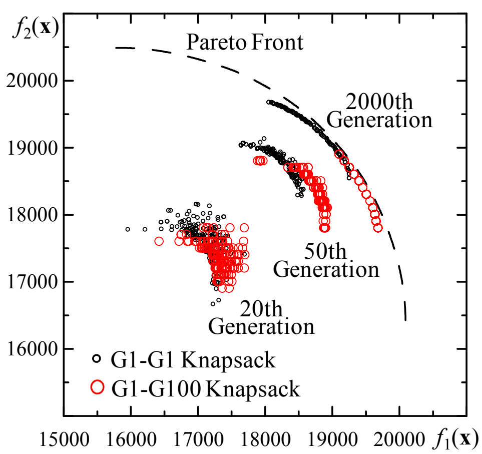

Figure 8. Experimental results of a single run of NSGA-II on the G1-G1 and G1-G100 knapsack problems.

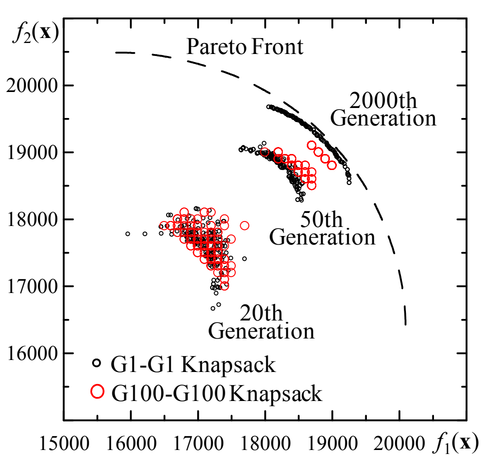

Figure 9. Experimental results of a single run of NSGA-II on the G1-G1 and G100-G100 knapsack problems.

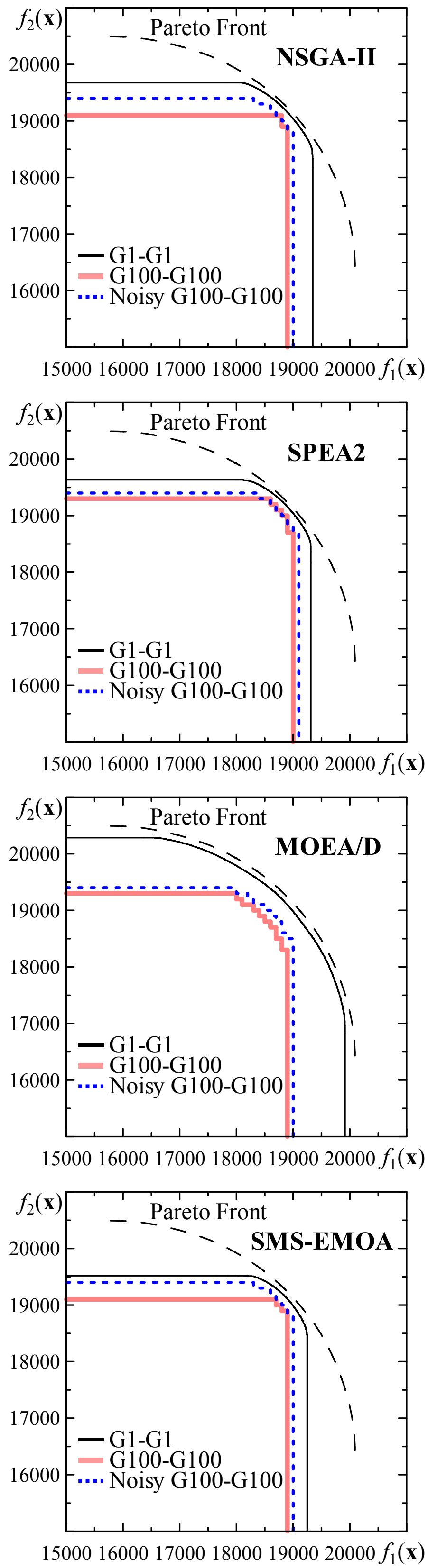

Figure 10. 50% attainment surface of each algorithm for the G1-G1, G100-G100 and noisy G100-G100 knapsack problems.

In Fig. 9 and Fig. 10, the use of the very coarse granularity G100 in $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ deteriorates the performance of all the four EMO algorithms (i.e., from results on G1-G1 to those on G100-G100).

In Fig. 10, the addition of a small random noise to $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ improves the performance of all the four EMO algorithms. A small random noise is not likely to make the search easier because it includes no information on the search direction. However, it actually improves the performance of the EMO algorithms in Fig. 10. This improvement may be due to the increase in the number of different non-dominated objective vectors.

3.3 Results on One-Max Zero-Max Problems

First, we show all solutions at the 2000th generation of a single run of NSGA-II on the linear G1-G1 and G10-G1 one-max zero-max problems in Fig. 11. The search by NSGA-II on the linear G10-G1 problem with the coarse granularity G10 in $f_1(\mathbf{x})$ is biased towards the top-left area in Fig. 11. For comparison, we show all solutions at the 2000th generation of a single run of NSGA-II on the concave G1-G1 and G10-G1 one-max zero-max problems in Fig. 12. This figure shows that the use of the coarse granularity G10 for the first objective of the concave G10-G1 problem biases the search of NSGA-II towards the top-left area (whereas the distribution of the different Pareto optimal objective vectors is sparse around the top-left area as shown in Fig. 5).

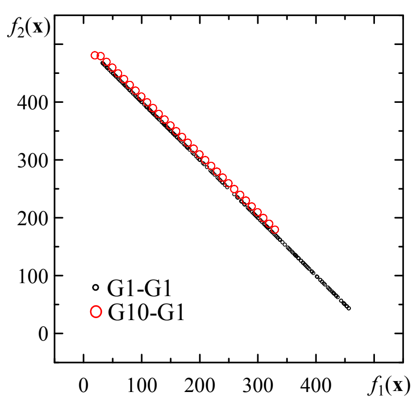

Figure 11. Solutions at the 2000th generation of NSGA-II on the linear G1-G1 and G10-G1 one-max zero-max problems.

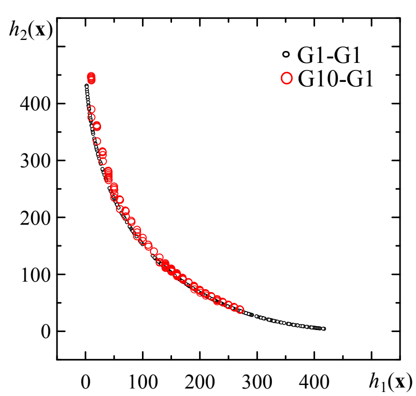

Figure 12. Solutions at the 2000th generation of NSGA-II on the concave G1-G1 and G10-G1 one-max zero-max problems.

In Figs. 13-15 in the next page, the 50% attainment surface over 100 runs of each EMO algorithm is shown for each test problem. As in Fig. 11, the search by each EMO algorithm in Fig. 13 is biased towards the top-left area by the use of the coarse granularity G10 in the first objective of the G10-G1 problem. In this case, we observe a clear improvement of the search towards the best solution of the second objective in the experimental results by SPEA2 and MOEA/D in Fig. 13. When we use the coarse granularity G10 in both objectives, the performance of all the four EMO algorithms is degraded in Figs. 13-15 (i.e., from results on G1-G1 to those on G10-G10).

In all experimental results in Figs. 13-15, we have the following two observations in common:

(i) The coarse granularity G10 in the first objective of the G10-G1 problem biases the search towards the top-left area.
(ii) The coarse granularity G10 in both objectives of the G10-G10 problem degrades the performance of the EMO algorithms.

That is, we have these two observations by all the four EMO algorithms for the three types of the one-max zero-max problems independently of the shape of their Pareto fronts (i.e., linear, convex and concave). This is interesting because the distribution of the Pareto optimal objective vectors is totally different between the convex and concave G10-G1 problems (see Fig. 4 and Fig. 5).

We also examine the performance of each EMO algorithm on the noisy versions of the three types of the G10-G10 problems. The addition of a small noise to each of the two objective functions improves the performances on the following cases:

NSGA-II on the concave G10-G10 problems,
SPEA2 on the convex and concave G10-G10 problems,
SMS-EMOA on all the three G10-G10 problems.

In the other cases (e.g., NSGA-II on the linear and convex G10-G10 problems), the addition of a small noise shows almost no effects on the performance of each EMO algorithm.

In Fig. 16, we show the experimental results of SMS-EMOA on the concave G1-G1, G10-G10 and noisy G10-G10 problems. This figure clearly shows that the addition of a small noise to each of the two objective functions improves the performance of SMS-EMOA on the concave G10-G10 one-max zero-max problem.

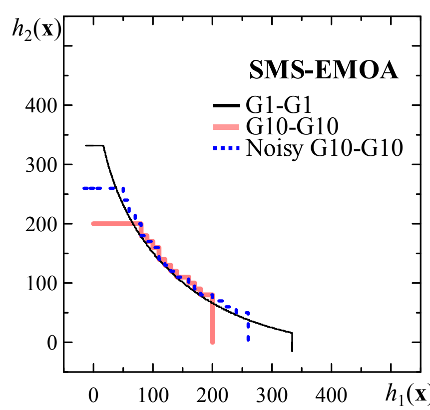

Figure 16. 50% attainment surface of SMS-EMOA for the concave G1-G1, G10-G10 and noisy G10-G10 one-max zero max problems.

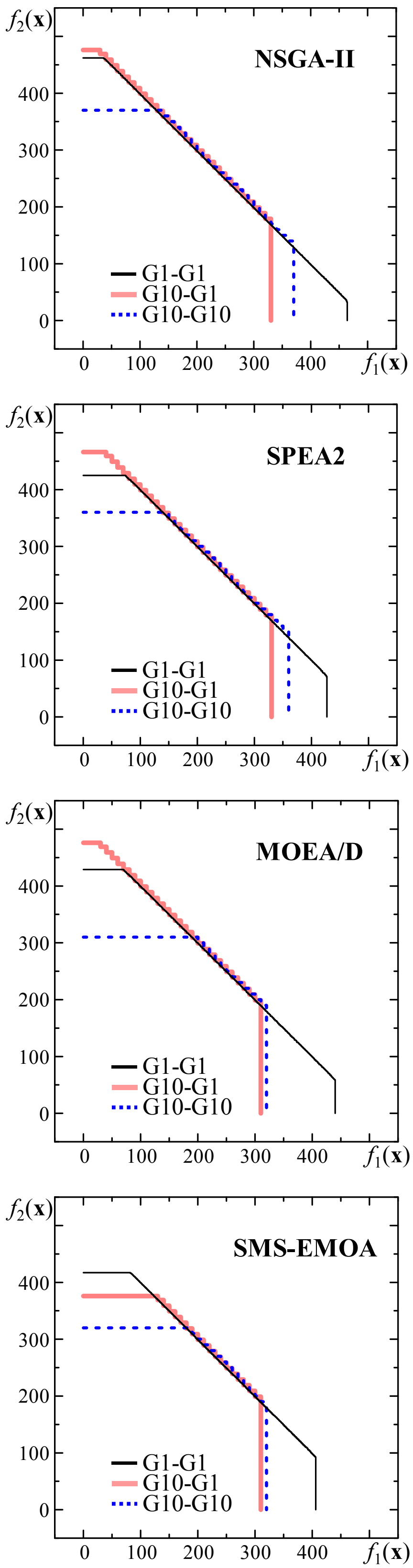

Figure 13. 50% attainment surface for the linear G1-G1, G10-G1, G10-G10.

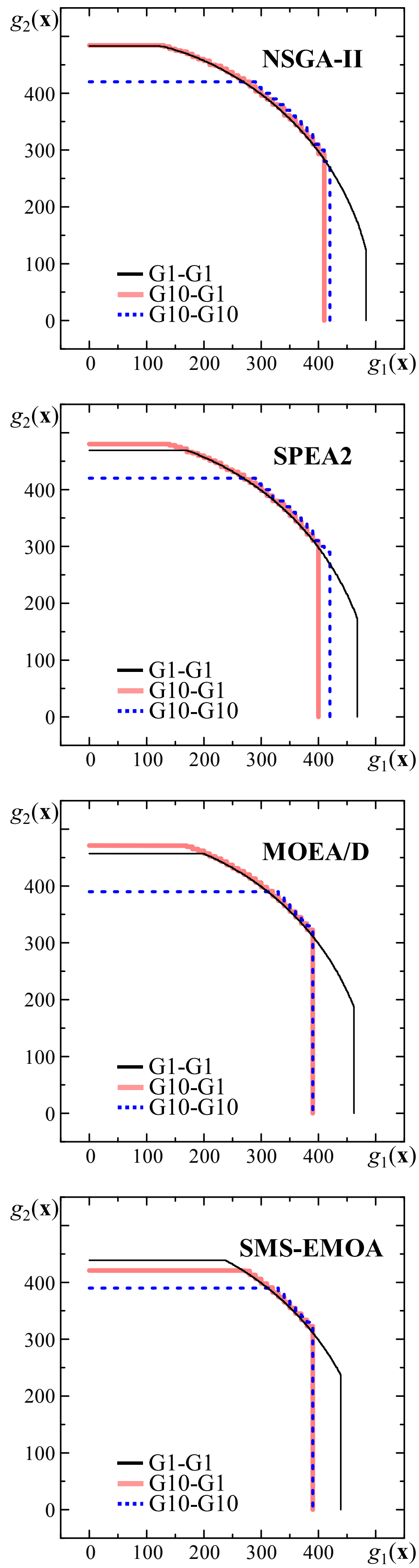

Figure 14. 50% attainment surface for the convex G1-G1, G10-G1, G10-G10.

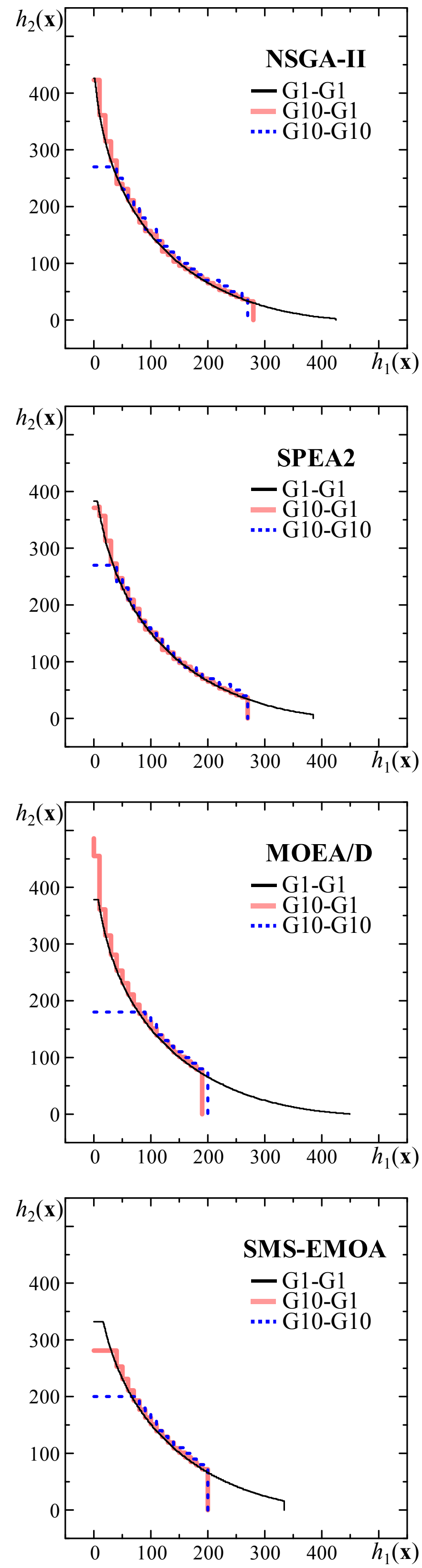

Figure 15. 50% attainment surface for the concave G1-G1, G10-G1, G10-G10.

4. CONCLUSIONS

In this paper, we examined the search behavior of the four well-known and frequently-used EMO algorithms (NSGA-II, SPEA2, MOEA/D and SMS-EMOA) on the two-objective combinatorial optimization problems with different granularities in their discrete objective functions. Through computational experiments, we clearly demonstrated that the difference in the granularities of discrete objective functions biased the search by those EMO algorithms towards a certain area of the objective space. To the best of our knowledge, the effect of the difference in the granularities of discrete objective functions has not been clearly demonstrated in the literature in the EMO community. We also demonstrated that the addition of a small noise to each discrete objective function with a coarse granularity (e.g., a random number in the interval [0, 1] to an objective function with the discretization interval of the width 100) improved the search ability of some EMO algorithms.

While these interesting observations were clearly demonstrated through computational experiments in this paper, the reason for such an interesting behavior of EMO algorithms was not clearly explained. A lot of empirical and theoretical studies may be needed to analyze and explain the behavior of EMO algorithms on multiobjective combinatorial optimization problems with different granularities in their discrete objective functions. While the effect of granularity difference in discrete objective functions on the behavior of EMO algorithms had not been discussed in the literature, a number of studies seem to be potentially related to our work in this paper. Among them are studies on the use of epsilon dominance [8], [10], the handling of overlapping solutions in the objective space [15], the evolutionary search with plateaus of constant fitness [11] and the landscape analysis based on neutral networks [9], [14].

5. REFERENCES

[1] Korte, B., and Vygen, J. *Combinatorial Optimization: Theory and Algorithms* (Fourth Edition). Springer, Berlin (2010).

[2] Beume, N., Naujoks, B., and Emmerich, M. SMS-EMOA: multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181, 3 (2007) 1653-1669.

[3] Blum, C., and Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys* 35, 3 (2003) 268-308.

[4] Coello, C. A. C., and Lamont, G. B. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore (2004).

[5] Coello, C. A. C., Van Veldhuizen, D. A., and Lamont, G. B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, Boston (2002).

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

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

[8] Giel, O., and Lehre, P. K. On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation* 18, 3(2010) 335-356.

[9] Harvey, I., and Thompson, A. Through the labyrinth evolution finds a way: A silicon ridge. *Lecture Notes in Computer Science 1259: ICES 1996*, 406-422.

[10] Horoba, C., and Neumann, F. Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization. *Proc. of 2008 Genetic and Evolutionary Computation Conference* (2008) 641-648.

[11] Jansen, T., and Wegener, I. Evolutionary algorithms - How to cope with plateaus of constant fitness and when to reject strings of the same fitness. *IEEE Trans. on Evolutionary Computation* 5, 6 (2001) 589-599.

[12] Jaszkiewicz, A. 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 (2002) 402-412.

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

[14] Katada, Y., and Ohkura, K. Analysis on topologies of fitness landscapes with both neutrality and ruggedness based on neutral networks. *Proc. of 2009 Genetic and Evolutionary Computation Conference* (2009) 1855-1856.

[15] Nojima, Y., Narukawa, K., Kaige, S., and Ishibuchi, H. Effects of removing overlapping solutions on the performance of the NSGA-II Algorithm. *Lecture Notes in Computer Science 3410: EMO 2005*, 341-354.

[16] Papadimitriou, C. H., and Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*. Dover, Mineola (1998).

[17] Wagner, T., Beume, N., and Naujoks, B. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. *Lecture Notes in Computer Science 4403: EMO 2007*, 742-756.

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

[19] Zitzler, E., Brockhoff, D., and Thiele, L. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. *Lecture Notes in Computer Science 4403: EMO 2007*, 862-876.

[20] Zitzler, E., and Künzli, S. Indicator-based selection in multiobjective search. *Lecture Notes in Computer Science 3242: PPSN VIII* (2004) 832-842.

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

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