An Empirical Study on the Handling of Overlapping Solutions in Evolutionary Multiobjective Optimization

Hisao Ishibuchi
Graduate School of Engineering
Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
+81-72-254-9350
hisaoi@cs.osakafu-u.ac.jp

Kaname Narukawa
Graduate School of Engineering
Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
+81-72-254-9351
kaname@ci.cs.osakafu-u.ac.jp

Yusuke Nojima
Graduate School of Engineering
Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
+81-72-254-9351
nojima@cs.osakafu-u.ac.jp

ABSTRACT

We focus on the handling of overlapping solutions in evolutionary multiobjective optimization (EMO) algorithms. First we show that there exist a large number of overlapping solutions in each population when EMO algorithms are applied to multiobjective combinatorial optimization problems with only a few objectives. Next we implement three strategies to handle overlapping solutions. One strategy is the removal of overlapping solutions in the objective space. In this strategy, overlapping solutions in the objective space are removed during the generation update phase except for only a single solution among them. As a result, each solution in the current population has a different location in the objective space. Another strategy is to remove overlapping solutions so that each solution in the current population has a different location in the decision space. The other strategy is the modification of Pareto ranking where overlapping solutions in the objective space are allocated to different fronts. As a result, each solution in each front has a different location in the objective space. Effects of each strategy on the performance of the NSGA-II algorithm are examined through computational experiments on multiobjective 0/1 knapsack problems, multiobjective flowshop scheduling problems, and multiobjective fuzzy rule selection problems.

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), multiobjective combinatorial optimization, diversity of non-dominated solutions, diversity-preserving strategies, NSGA-II.

1. INTRODUCTION

Since Schaffer's pioneering work [13], the design of evolutionary multiobjective optimization (EMO) algorithms has been discussed to find a variety of well-distributed Pareto-optimal or near Pareto-optimal solutions (e.g., see Coello et al. [1] and Deb [2]). The handling of overlapping solutions, however, has not been discussed explicitly in many studies. This is mainly because the performance of EMO algorithms has been evaluated through computational experiments on multiobjective optimization problems with continuous decision variables. Since EMO algorithms usually have some sort of diversity-preserving mechanisms, many overlapping solutions are not likely to exist in each population when they are applied to multiobjective optimization problems with continuous decision variables and/or many objective functions. On the other hand, the handling of overlapping solutions becomes an important issue in the application of EMO algorithms to multiobjective combinatorial optimization problems with only a few objective functions. In such an application, there may exist a large number of overlapping solutions in each population as we will show further below through computational experiments on some test problems.

In this paper, we implement three strategies to handle overlapping solutions, and examine their effects on the performance of EMO algorithms. As a representative EMO algorithm, we use the NSGA-II algorithm of Deb et al. [3] because it is one of the most frequently used and well-known EMO algorithms in recent EMO studies. One strategy is the removal of overlapping solutions in the objective space. Overlapping solutions with the same location in the objective space are removed except for a randomly chosen single solution among them. As a result, each solution in the current population has a different location in the objective space. Another strategy is the removal of overlapping solutions in the decision space so that each solution in the current population has a different location in the decision space. It should be noted that overlapping solutions in the objective space are not necessarily the same solution (i.e., they are not necessarily overlapping with each other in the decision space). Thus there may exist some overlapping solutions in the objective space when we use the second strategy. The other strategy is the modification of Pareto ranking where overlapping solutions in the objective space are allocated to different fronts. As a result, each solution in each front has a different location in the objective space. Effects of each strategy on the performance of the NSGA-II algorithm are examined through computational experiments on multiobjective 0/1 knapsack problems, multiobjective flowshop scheduling problems, and multiobjective fuzzy rule selection problems.

2. OVERLAPPING SOLUTIONS

Before discussing the handling of overlapping solutions, we examine whether each population has many overlapping solutions in the application of the NSGA-II algorithm to some test problems.

2.1 Test Problems

Our computational experiments are performed on some test problems: a two-objective test problem called Min-Ex in Deb [2], ZDT1, ZDT2, and ZDT3 of Zitzler et al. [14], multiobjective 0/1 knapsack problems of Zitzler & Thiele [15], multiobjective flowshop scheduling problems of Ishibuchi et al. [8], and multiobjective fuzzy rule selection problems of Ishibuchi & Yamamoto [7]. Here these test problems are briefly explained.

Deb [2] used the following two-objective test problem with two continuous variables called Min-Ex to illustrate characteristic features of a number of EMO algorithms:

$$\text{Min-Ex: Minimize } f_1(\mathbf{x}) = x_1 \text{ and } f_2(\mathbf{x}) = (1 + x_2)/x_1, \quad (1)$$

$$\text{subject to } 0.1 \le x_1 \le 1 \text{ and } 0 \le x_2 \le 5. \quad (2)$$

The decision space of this problem is $[0.1,1] \times [0,5]$. We represent each variable by a binary string of length 30 using standard binary coding.

Zitzler et al. [14] framed six test problems (ZDT1 to ZDT6) to examine the performance of a number of EMO algorithms. We use the first three test problems (ZDT1 to ZDT3), which have 30 continuous variables in the unit interval [0, 1]. That is, the decision space of these test problems is the 30-dimensional unit hyper-cube $[0, 1]^{30}$. All the six test problems in [14] can be written in the following form:

$$\text{Minimize } f_1(\mathbf{x}) \text{ and } f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x})). \quad (3)$$

In all the first three test problems (ZDT1 to ZDT3), $f_1(\mathbf{x})$ and $g(\mathbf{x})$ are defined as follows:

$$f_1(\mathbf{x}) = x_1 \text{ and } g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=1}^{n} x_i, \quad (4)$$

where n is the number of decision variables (i.e., 30). The first three test problems are different from one another in the definition of the function $h(f_1, g)$ as follows:

$$\text{ZDT1: } h(f_1, g) = 1 - \sqrt{f_1 / g}, \quad (5)$$

$$\text{ZDT2: } h(f_1, g) = 1 - (f_1 / g)^2, \quad (6)$$

$$\text{ZDT3: } h(f_1, g) = 1 - \sqrt{f_1 / g} - (f_1 / g) \cdot \sin(10\pi f_1). \quad (7)$$

Zitzler & Thiele [15] used nine multiobjective 0/1 knapsack problems, each of which has two, three or four objectives and 250, 500 or 750 items. Each test problem with k knapsacks (i.e., k objectives and k constraints) and n items can be written as

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})), \quad (8)$$

$$\text{subject to } \sum_{j=1}^{n} w_{ij} x_j \le c_i, \quad i = 1, 2, ..., k, \quad (9)$$

$$\text{where } f_i(\mathbf{x}) = \sum_{j=1}^{n} p_{ij} x_j, \quad i = 1, 2, ..., k. \quad (10)$$

In this formulation, $\mathbf{x}$ is an n-dimensional binary vector, p_{ij} is the profit of item j according to knapsack i, w_{ij} is the weight of item j according to knapsack i, and c_i is the capacity of knapsack i. Each solution $\mathbf{x}$ is handled as a binary string of length n. We denote the k-objective n-item test problem as the k-n knapsack problem. Multiobjective 0/1 knapsack problems have been used in many studies to examine the performance of EMO algorithms (e.g., Jaszkiewicz [9], [10], Knowles & Corne [11], Mumford [12], and Zydallis & Lamont [16]).

When an EMO algorithm is applied to the multiobjective 0/1 knapsack problem in (8)-(10), genetic operations often generate infeasible solutions that do not satisfy the constraint conditions in (9). We use a repair method based on a maximum profit/weight ratio as suggested by Zitzler & Thiele [15]. When an infeasible solution is generated, a feasible solution is created by removing items (i.e., by changing the corresponding values in the binary string $\mathbf{x}$ from 1 to 0) in the ascending order of the maximum profit/weight ratio defined as follows:

$$q_j = \max\{p_{ij} / w_{ij} \mid i = 1, 2, ..., k\}, \quad j = 1, 2, ..., n. \quad (11)$$

In Ishibuchi et al. [8], multiobjective flowshop scheduling problems were used to examine the performance of some EMO and memetic EMO algorithms. Each solution of a flowshop scheduling problem with n jobs is represented by a permutation of the given n jobs $\{J_1, J_2, ..., J_n\}$. Two-objective test problems in [4], [8] are written as

$$\text{Minimize } f_1(\mathbf{x}) = \max\{C_i \mid i = 1, 2, ..., n\}, \quad (12)$$

$$\text{Minimize } f_2(\mathbf{x}) = \max\{\max\{C_i - d_i, 0\} \mid i = 1, 2, ..., n\}, \quad (13)$$

where C_i and d_i are the completion time and the due-date of the i-th job J_i, respectively. The first objective is to minimize the makespan (i.e., the maximum completion time) while the second objective is to minimize the maximum tardiness. Three-objective flowshop scheduling problems in [4], [8] has the following third objective in addition to the two objectives in (12)-(13):

$$\text{Minimize } f_3(\mathbf{x}) = \sum_{i=1}^{n} C_i. \quad (14)$$

Each flowshop scheduling problem in [8] has 20 machines and 20, 40, 60 or 80 jobs. We denote the k-objective test problem with n jobs as the k-n scheduling problem.

As test problems, we also use multiobjective fuzzy rule selection problems of Ishibuchi & Yamamoto [6], [7]. They employed EMO and memetic EMO algorithms for multiobjective design of fuzzy rule-based classification systems in the following manner. First a prespecified number of candidate fuzzy rules were generated from training patterns for each class. For details of fuzzy rule generation, see the textbook on fuzzy data mining by Ishibuchi et al. [5]. Then non-dominated rule sets were found from the generated candidate rules with respect to the following three objectives:

$$\text{Maximize } f_1(\mathbf{x}), \text{ minimize } f_2(\mathbf{x}), \text{ and minimize } f_3(\mathbf{x}), \quad (15)$$

where $f_1(\mathbf{x})$ is the number of correctly classified training patterns by selected rules, $f_2(\mathbf{x})$ is the number of selected rules, and $f_3(\mathbf{x})$ is the total number of antecedent conditions of selected rules.

Let N be the total number of generated candidate rules (i.e., N/M is the number of generated candidate rules for each class where M is the number of classes). Then each solution of multiobjective rule selection problems is represented by a binary string of length N. In addition to the three-objective fuzzy rule selection problems, we also use two-objective problems with the first two-objectives:

$$\text{Maximize } f_1(\mathbf{x}) \text{, and minimize } f_2(\mathbf{x}) . \quad (16)$$

We denote the k-objective test problem with N candidate fuzzy rules as the k-N rule selection problem.

2.2 Results on Function Optimization

In this subsection, we report our experimental results on the four multiobjective function optimization problems with continuous decision variables: Min-Ex, ZDT1, ZDT2, and ZDT3. The NSGA-II algorithm was applied to each test problem 20 times using the following parameter specifications:

Population size: 200,
Crossover probability: 0.9 (One-point crossover),
Mutation probability: $1/N$ (N is the string length),
Stopping condition: 5000 generations.

In Figure 1, we show the average number of overlapping solutions in the objective space at each generation for each test problem. From this figure, we can see that the average number of overlapping solutions was always smaller than half of the population size in all the four test problems (e.g., it was about ten in the case of Min-Ex in Figure 1).

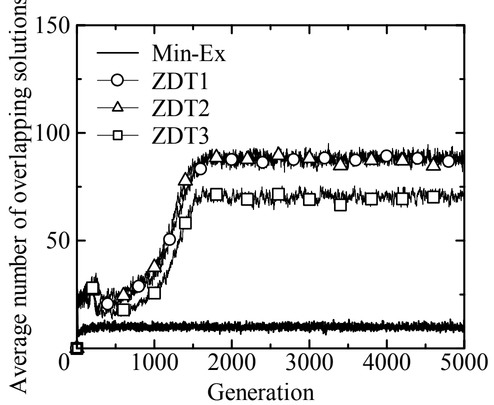

Figure 1. Average number of overlapping solutions in the objective space at each generation.

In Figure 2, we also show the average number of different solutions in the objective space. From this figure, we can see that each population had a large number of different solutions (e.g., at least 150 different solutions on average) during the execution of the NSGA-II algorithm on each of the four test problems. Here we briefly explain the relation between the number of overlapping solutions in Figure 1 and the number of different solutions in Figure 2. Let us assume that the current population with N_{pop} solutions includes N_{overlap} overlapping solutions. In this case, the number of different solutions can be $N_{\text{pop}} - N_{\text{overlap}} + 1$ (when all the overlapping solutions have the same location in the objective space), $N_{\text{pop}} - N_{\text{overlap}}/2$ (when each pair of overlapping solutions have a different location in the objective space), or an integer between these two extreme cases. Thus we can not calculate the number of different solutions from the number of overlapping solutions. This is why we reported experimental results with respect to these two statistics.

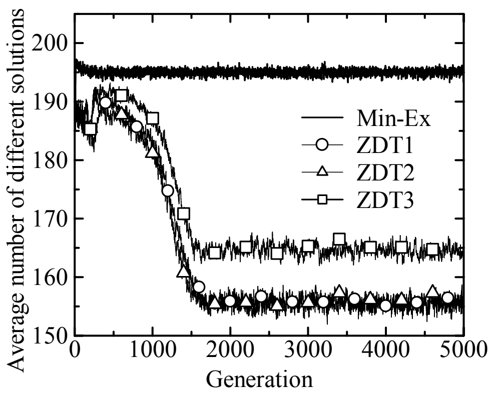

Figure 2. Average number of different solutions in the objective space at each generation.

Experimental results in Figure 1 and Figure 2 suggest that the existence of overlapping solutions is not a serious issue in the application of EMO algorithms to multiobjective function optimization problems with continuous decision variables. This may be the reason why this issue has not been discussed in many studies on EMO algorithms.

2.3 Results on Knapsack Problems

We applied the NSGA-II algorithm to the three two-objective knapsack problems (i.e., 2-250, 2-500, 2-750 knapsack problems) 20 times using the following parameter specifications:

Population size: 150 (2-250), 200 (2-500), 250 (2-750),
Crossover probability: 0.8 (One-point crossover),
Mutation probability: $1/N$ (N is the string length),
Stopping condition: 5000 generations.

Figure 3 shows the average number of overlapping solutions in the objective space at each generation for each test problem. From this figure, we can see that the number of overlapping solutions increased rapidly in early generations, then gradually decreased in late generations. We also show the number of different solutions in Figure 4. The number of different solutions decreased rapidly about by half during the first 1000 generations, then increased gradually in late generations. Figure 3 and Figure 4 suggest that the existence of overlapping solutions may have some effects on the performance of the NSGA-II algorithm on the two-objective knapsack problems.

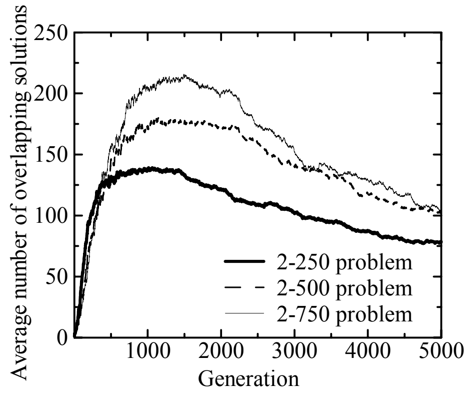

Figure 3. Average number of overlapping solutions during the execution of NSGA-II on two-objective knapsack problems.

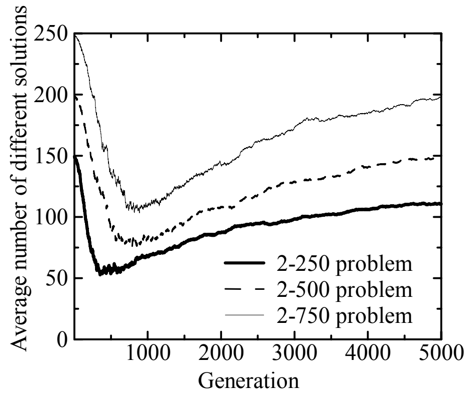

Figure 4. Average number of different solutions during the execution of NSGA-II on two-objective knapsack problems.

In order to examine the effect of the population size on the number of different solutions, we applied the NSGA-II algorithm to the 2-500 knapsack problem 20 times using the following three specifications of the population size: 200, 400, and 600. Experimental results are shown in Figure 5 in the same manner as Figure 4. From this figure, we can see that there were not large differences in the number of different solutions in late generations while there were large differences in the population size among the three cases in Figure 5. This observation suggests that the use of a large population (e.g., 600 solutions) does not necessarily have a positive effect on the performance of the NSGA-II algorithm with respect to the diversity of solutions while it significantly increases the computational load. That is, the increase in the population size is not likely to lead directly to the increase in the number of obtained non-dominated solutions.

We also examined the effect of the number of objectives on the number of overlapping solutions. We applied the NSGA-II algorithm to the 2-500, 3-500 and 4-500 knapsack problems 20 times using the following specifications of the population size.

Population size: 200 (2-500), 250 (3-500), 300 (4-500).

The other parameter values were specified in the same manner as the previous computational experiments on the two-objective knapsack problems. We show the average number of overlapping solutions during the first 500 generations in Figure 6 where the results on the 2-500 knapsack problem are the same as Figure 3. We can see from Figure 6 that many overlapping solutions did not exist during the execution of the NSGA-II algorithm on the knapsack problems with three or more objectives.

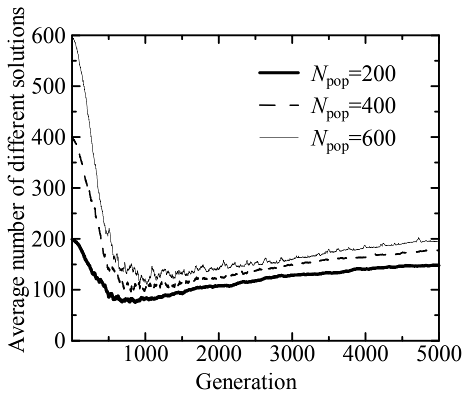

Figure 5. Average number of different solutions during the execution of NSGA-II on the 2-500 knapsack problem.

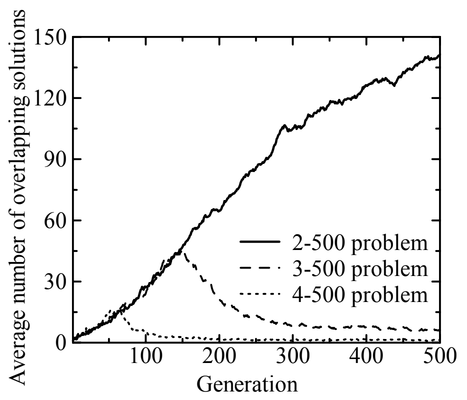

Figure 6. Average number of overlapping solutions during the execution of NSGA-II on knapsack problems with two, three and four objectives.

2.4 Results on Scheduling Problems

We applied the NSGA-II algorithm to the four two-objective flowshop scheduling problems with 20, 40, 60, and 80 jobs (i.e., 2-20, 2-40, 2-60, and 2-80 scheduling problems) 20 times using the following parameter specifications:

Population size: 120 (2-20), 140 (2-40), 160 (2-60), 180 (2-80),
Crossover probability: 0.9 (Two-point order crossover),
Mutation probability: 0.8 per string (Shift mutation),
Stopping condition: 5000 generations.

In Figure 7, we show the average number of different solutions in the objective space. From Figure 7, we can see that each population had only 20-30 different solutions while the population size was 120-180. This means that there existed many overlapping solutions during the execution of the NSGA-II algorithm on the two-objective flowshop scheduling problems.

In the same manner, we also applied the NSGA-II algorithm to the four three-objective flowshop scheduling problems (i.e., 3-20, 3-40, 3-60, and 3-80) 20 times. Experimental results are summarized in Figure 8 in the same manner as Figure 7. While the number of different solutions was very small in the case of the two-objective flowshop scheduling problems in Figure 7, each population had a large number of different solutions in the case of three objectives in Figure 8.

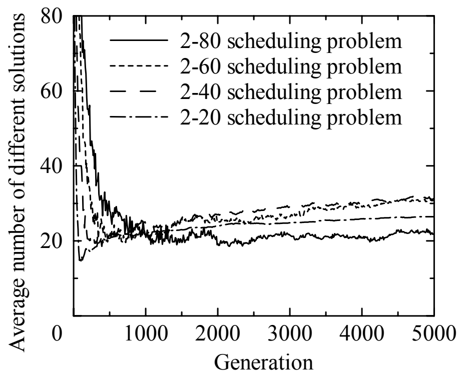

Figure 7. Average number of different solutions during the execution of NSGA-II on two-objective scheduling problems.

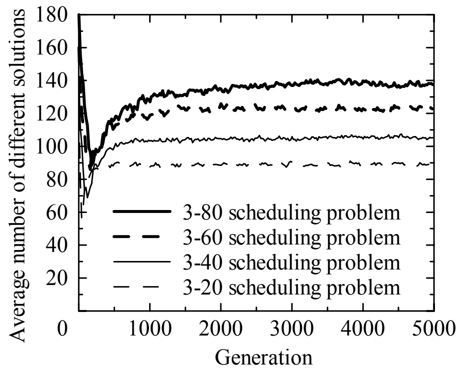

Figure 8. Average number of different solutions during the execution of NSGA-II on three-objective scheduling problems.

2.5 Results on Fuzzy Rule Selection Problems

In computational experiments on fuzzy rule selection problems, we first generated a prespecified number of candidate fuzzy rules for each of the three classes of the sonar data set in the UC Irvine Machine Learning Repository. The number of candidate fuzzy rules for each class was specified as 50, 100, 200, and 500 (i.e., 150, 300, 600, and 1500 in total). Then the NSGA-II algorithm was applied to two-objective and three-objective fuzzy rule selection problems ten times. We used the following parameter specifications for the two-objective problems:

Population size:
120 (2-150), 140 (2-300), 160 (2-600), 180 (2-1500),
Crossover probability: 0.8 (One-point crossover),
Mutation probability:
0.1 for the mutation "1 $\rightarrow$ 0",
1/N for the mutation "0 $\rightarrow$ 1" (N is the string length),
Stopping condition: 5000 generations.

We biased the mutation probability to decrease efficiently the number of fuzzy rules in each string (for detail, see [5]). In the same manner as the case of two objectives, we also applied the NSGA-II algorithm to the three-objective fuzzy rule selection problems ten times. Experimental results are summarized in Figure 9 and Figure 10.

From Figure 9, we can see that the number of different solutions was very small (i.e., about 5-10) in the case of two-objective fuzzy rule selection while the population size was 120-180. The number of different solutions slightly increased in the case of three objectives in Figure 10 but it was still very small.

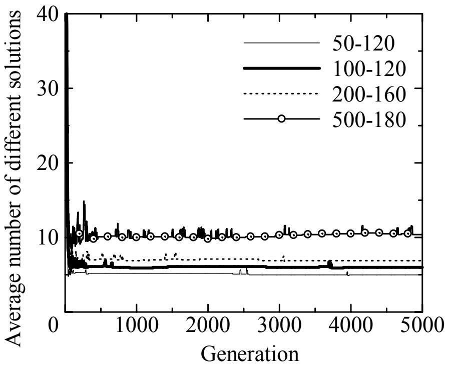

Figure 9. Average number of different solutions during the execution of NSGA-II for two-objective rule selection.

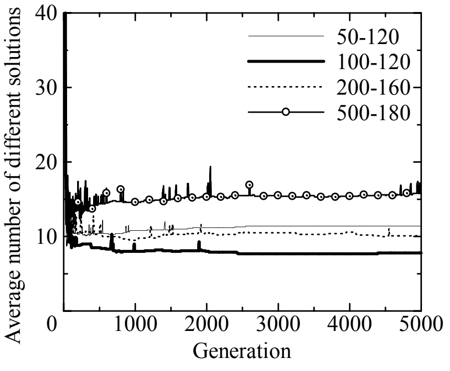

Figure 10. Average number of different solutions during the execution of NSGA-II on three-objective rule selection.

3. STRATEGIES FOR THE HANDLING OF OVERLAPPING SOLUTIONS

We have already demonstrated that there existed a large number of overlapping solutions during the execution of the NSGA-II algorithm on some multiobjective combinatorial optimization problems. In this section, we examine three strategies for the handling of overlapping solutions. Each strategy is incorporated into the NSGA-II algorithm to examine its effectiveness.

3.1 Three Strategies

We implemented the following three strategies and incorporated each of them into the NSGA-II algorithm.

Removal in the Objective Space: In this strategy, overlapping solutions in the objective space are removed during the generation update phase of the NSGA-II algorithm except for a single randomly chosen solution among them. As a result, each solution in the next population has a different location in the objective space. The point in the implementation of this strategy is to generate an initial population using different solutions in the objective space. Since the next population is constructed from the current population and the newly generated offspring population in the NSGA-II algorithm, it is always possible to construct the next population with no overlapping solutions when the initial population has no overlapping solutions.

Removal in the Decision Space: In this strategy, overlapping solutions in the decision space are removed except for a single solution among them. As a result, each solution in the next population has a different location in the decision space. This strategy can be implemented in the same manner as the removal strategy in the objective space.

Modification of Pareto Ranking: In the NSGA-II algorithm, solutions in the current population are divided into some fronts based on Pareto ranking in order to evaluate their fitness. First all the non-dominated solutions in the current population are allocated to the first front and tentatively removed from the current population. Next all the non-dominated solutions in the reduced current population are allocated to the second front and tentatively removed from the reduced current population. In this manner, all the solutions are divided into a number of different fronts. Since overlapping solutions are non-dominated with one another, all the overlapping solutions with the same location in the objective space are allocated to the same front. We modify this procedure as follows: Only a single solution among the overlapping solutions can be allocated to the same front. This modification of Pareto ranking is used not only in the selection phase to choose parent solutions for recombination but also in the generation update phase to construct the next population from the current population and the offspring population.

3.2 Experimental Results

We applied the NSGA-II algorithm with each strategy for the handling of overlapping solutions to two-objective 0/1 knapsack problems, flowshop scheduling problems with two and three objective, and three-objective fuzzy rule selection problems. In computational experiments on fuzzy rule selection, we used six data sets (i.e., Breast W, Diabetes, Glass, Heart C, Sonar, and Wine) from the UC Irvine Machine Learning Repository. First 300 candidate rules were generated for each class in each data set. Then the modified NSGA-II algorithm was applied to the three-objective rule selection problem.

Average results on 20 independent runs on each of the two-objective 0/1 knapsack problems (i.e., 2-250, 2-500, and 2-750 knapsack problems) are summarized in Table 1 where the average CPU time (in seconds) over all experiments are also shown. In Table 1, different non-dominated solutions are counted in the objective space. The largest number in each row is highlighted by boldface in Table 1. From this table, we can see that the number of obtained solutions was increased by the removal of overlapping solutions. The modification of Pareto ranking, however, did not increase the number of obtained non-dominated solutions.

Average results on the 2-250 and 2-500 knapsack problems are also shown in the form of 50% attainment surface together with the true Pareto front [15] in Figure 11 and Figure 12, respectively. An interesting observation in Figure 11 and Figure 12 is that the removal of overlapping solutions improved the convergence of solutions to the Pareto front while it increased the number of obtained non-dominated solutions. This may be explained as follows. When the current population includes multiple copies of the same solution, the chance of recombining the same solution is larger than the case of no overlapping solutions. Thus the removal of overlapping solutions has the same effect as increasing the crossover probability. The removal of overlapping solutions may improve the convergence of solutions to the Pareto front through the same effect as increasing the crossover probability.

Table 1. Average number of obtained non-dominated solutions for the two-objective 0/1 knapsack problems.

Problem	Original NSGA-II	Removal (objective)	Removal (decision)	Modified ranking
2-250	50.35	58.80	**61.65**	50.75
2-500	52.95	57.55	**59.75**	50.55
2-750	69.90	**71.20**	69.75	59.95
Average	57.73	62.52	**63.72**	53.75
CPU time	113.8	123.95	148.77	124.63

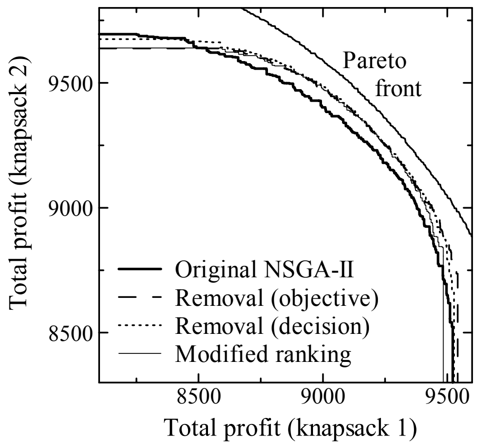

Figure 11. 50% attainment surface by each algorithm for the 2-250 knapsack problem.

In order to examine the validity of this explanation, we evaluated the performance of the original NSGA-II algorithm on the 2-500 knapsack problem using three specifications of the crossover probability: 0.8, 0.9 and 1.0. Experimental results are shown in Figure 13. From this figure, we can see that the convergence of solutions was improved by increasing the crossover probability.

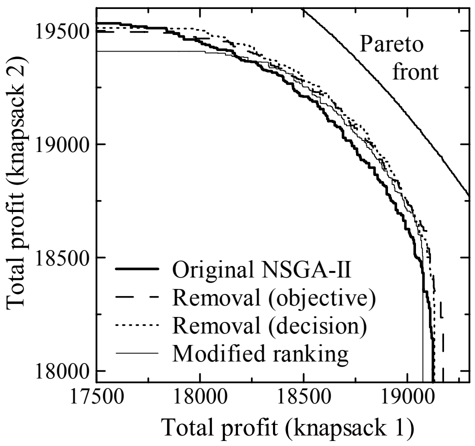

Figure 12. 50% attainment surface by each algorithm for the 2-500 knapsack problem.

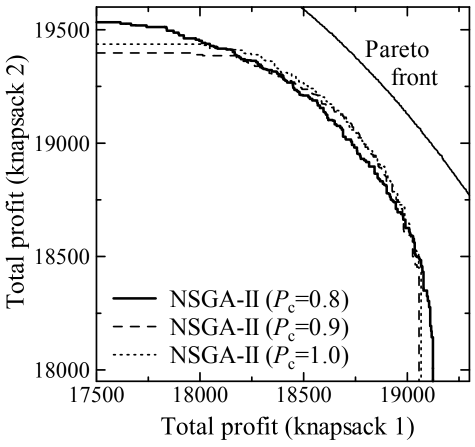

Figure 13. 50% attainment surface by the NSGA-II algorithm with different specifications of the crossover probability.

In Figure 14, we compare the two removal strategies with the original NSGA-II algorithm with higher crossover probabilities. From this figure, we can see that better results were obtained with respect to both the convergence to the Pareto front and the diversity of obtained solutions by removing overlapping solutions than the original NSGA-II algorithm.

Next we show experimental results on multiobjective flowshop scheduling problems in Table 2 in the same manner as Table 1. That is, the average number of obtained non-dominated solutions in the objective space is shown in Table 2. In this table, all the three strategies for the handling of overlapping solutions increased the number of obtained non-dominated solutions.

Finally we show experimental results on the three-objective fuzzy rule selection problems in Table 3 and Table 4 where the average number of different non-dominated solutions in the objective and decision spaces is summarized, respectively.

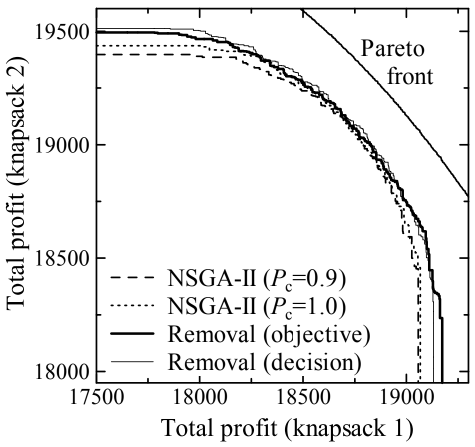

Figure 14. Comparison between the NSGA-II algorithm with higher crossover probabilities and the modified NSGA-II algorithm with the removal of overlapping solutions.

Table 2. Average number of obtained non-dominated solutions for the multiobjective flowshop scheduling problems.

Problem	Original NSGA-II	Removal (objective)	Removal (decision)	Modified ranking
2-20	26.5	31.3	**31.5**	27.0
2-40	31.4	**38.2**	36.8	32.4
2-60	30.8	33.5	27.3	**36.1**
2-80	22.0	**28.4**	21.5	23.8
3-20	91.1	121.0	121.1	**121.2**
3-40	108.1	**141.3**	140.6	140.7
3-60	123.4	160.4	**160.8**	**160.8**
3-80	139.1	180.1	**180.3**	179.5
Average	71.5	**91.8**	90.0	90.2

Table 3. Average number of different solutions in the objective space of three-objective rule selection problems.

Problem	Original NSGA-II	Removal (objective)	Removal (decision)	Modified ranking
Breast W	12.8	**14.0**	13.2	13.4
Diabetes	15.2	**18.3**	16.2	16.8
Glass	31.6	**32.9**	31.5	31.6
Heart C	**39.4**	38.9	39.2	38.6
Sonar	12.2	**14.2**	13.2	13.0
Wine	12.9	**13.7**	12.3	13.4
Average	20.7	**22.0**	20.9	21.1

Table 4. Average number of different solutions in the decision space of three-objective rule selection problems.

Problem	Original NSGA-II	Removal (Objective)	Removal (Decision)	Modified Ranking
Breast W	13.9	14.0	**51.8**	13.4
Diabetes	15.9	18.3	**30.0**	16.8
Glass	34.2	32.9	**177.7**	34.2
Heart C	92.6	38.9	**200.7**	38.6
Sonar	13.1	14.2	**32.9**	13.0
Wine	13.9	13.7	**50.0**	13.4
Average	30.6	22.0	**90.5**	21.6

4. CONCLUSIONS

In this paper, we first examined whether a large number of overlapping solutions were included in each population through computational experiments on some multiobjective test problems using the NSGA-II algorithm. Experimental results showed that there existed a large number of overlapping solutions when the NSGA-II algorithm was applied to two-objective 0/1 knapsack problems, two-objective flowshop scheduling problems, and fuzzy rule selection problems with two or three objectives. On the other hand, the number of overlapping solutions was not large in the application to multiobjective function optimization problems with continuous decision variables (i.e., Min-EX, ZDT1, ZDT2 and ZDT3), multiobjective 0/1 knapsack problems with three or more objectives, and three-objective flowshop scheduling problems. Then we implemented three strategies for the handling of overlapping solutions. Their effectiveness was examined through computational experiments. Experimental results showed that each strategy increased the number of obtained non-dominated solutions in many cases. For example, the removal of overlapping solutions in the objective or decision space increased the number of obtained solutions in the corresponding space (see Table 3 and Table 4). An interesting observation in experimental results on two-objective knapsack problems was that the removal of overlapping solutions improved the convergence of solutions to the Pareto front (see Figure 11 and Figure 12).

5. ACKNOWLEDGMENTS

The authors would like to thank Ms. Shiori Kaige for her help with computational experiments. This work was partially supported by Kayamori Foundation of Information Science Advancement, and Japan Society for the Promotion of Science (JSPS) through Grand-in-Aid for Scientific Research (B): KAKENHI (14380194).

6. REFERENCES

[1] Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, Boston, MA, 2002.

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

[3] 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 (April 2002) 182-197.

[4] Ishibuchi, H. and Murata, T. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews 28*, 3 (August 1998) 392-403.

[5] Ishibuchi, H., Nakashima, T., and Nii, M. *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Springer, Berlin, 2004.

[6] Ishibuchi, H., and Yamamoto, T. Effects of three-objective genetic rule selection on the generalization ability of fuzzy rule-based systems. *Lecture Notes in Computer Science 2632: Evolutionary Multi-Criterion Optimization*, Springer, Berlin (April 2003) 608-622.

[7] Ishibuchi, H., and Yamamoto, T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems 141*, 1 (January 2004) 59-88.

[8] Ishibuchi, H., Yoshida, T., and Murata, T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans. on Evolutionary Computation 7*, 2 (April 2003) 204-223.

[9] Jaszkiewicz, A. Comparison of local search-based metaheuristics on the multiple objective knapsack problem. *Foundations of Computing and Decision Sciences 26*, 1 (2001) 99-120.

[10] 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 (August 2002) 402-412.

[11] Knowles, J. D., and Corne, D. W. A comparison of diverse approaches to memetic multiobjective combinatorial optimization. *Proc. of 2000 Genetic and Evolutionary Computation Conference Workshop Program: WOMA I* (Las Vegas, NV, July 8-12, 2000) 103-108.

[12] Mumford, C. L. Comparing representations and recombination operators for the multi-objective 0/1 knapsack problem. *Proc. of 2003 Congress on Evolutionary Computation* (Canberra, Australia, December 8-12, 2003) 854-861.

[13] Schaffer, J. D. Multiple objective optimization with vector evaluated genetic algorithms. *Proc. of 1st International Conference on Genetic Algorithms and Their Applications* (Pittsburgh, PA, July 24-26, 1985) 93-100.

[14] Zitzler, E., Deb, K., and Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation 8*, 2 (Summer 2000) 125-148.

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

[16] Zydallis, J. B., and Lamont, G. B. Explicit building-block multiobjective evolutionary algorithms for NPC problems. *Proc. of 2003 Congress on Evolutionary Computation* (Canberra, Australia, December 8-12, 2003) 2685-2695.